Team Awesome Optimized Digital Signal Processor

Mark Cheung
ECE 3663 – Spring 2013
University of Virginia
mc5ah@virginia.edu

Michael Recachinas
ECE 3663 – Spring 2013
University of Virginia
mgr3yp@virginia.edu

Piyapath Siratarnsophon
ECE 3663 – Spring 2013
University of Virginia
ps3an@virginia.edu

Ta Wei (David) Yi
ECE 3663 – Spring 2013
University of Virginia
ty6py@virginia.edu

ABSTRACT
The arithmetic logic unit (ALU) is the fundamental part of the central processing unit (CPU) in microprocessors. Digital Signal Processors (DSP) are specialized microprocessors that compute many mathematical operations quickly. With transistors' sizes quickly approaching 20 nm, heavy focus is placed upon total area, delay, and power consumption. Various techniques have been implemented for optimizing ALUs and DSPs. In this paper we discuss our innovative design optimized for speed and simulation results of the DSP.

Categories and Subject Descriptors
B.7 [**Integrated Circuits**]: Types and Design Styles—*Microprocessors, VLSI (very large scale integration)*; C.1 [**Processor Architectures**]: General

General Terms
Optimization

Keywords
ALU, CMOS, DSP, comparator, logical effort

1. INTRODUCTION
Designs and simulations took place using FreePDK's 45nm technology. Hoping to win the contract from Itty Bitty Logic Co. (IBL), we built the DSP following IBL's design constraints, maximizing the speed. This paper describes our design choices and innovations. To prove success in our design, transistor level hierarchical netlist of the DSP and the Cadence simulations are attached.

2. DESIGN DESCRIPTION
The DSP consists of an ALU with the functions NOP, ADD, 2COMP, SHIFT, AND, OR, PASS A, and Comparator (Table 1). A 3-bit control value chooses the operation, registers placed before and after the 16-bit inputs and output hold the values, and buffers prevent the signal from weakening. Figure 1 illustrates a high-level schematic of the requested DSP.

Table 1: ALU Functions and Descriptions

ALU Function	Description	Control
`NOP`	No change at Out	000
`ADD`	Out = $A + B$	001
`2COMP`	Out = Two's Complement of A	010
`SHIFT`	Out = $A << B$ (B signed)	011
`AND`	Out = $A\&B$	100
`OR`	Out = $A\|B$	101
`NOP/Pass A`	Out = A	110
`Comparator`	Out = (A >= B ? 1 : 0)	111

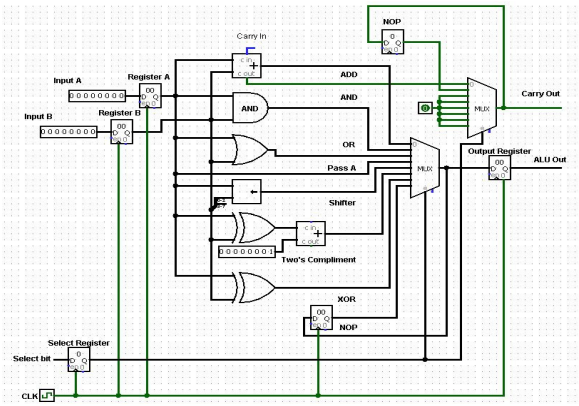

Figure 1: IBL Digital Signal Processor Topology (*Logisim*)

2.1 V_{dd} Value
We compared the V_{DD} value by computing delay$^2\cdot$ power since the area isn't affected by change in voltage. As seen in Table 2, 1.0 V provides the best metric.

2.2 Kogge-Stone Adder
Originally, we considered a mirror adder for the design of the final ALU adder. However, after testing each available component with Ocean, we found that the mirror adder had

Table 2: $D^2 \cdot P$ of Same Processor with Different V_{dd}

V_{dd}	Delay (ps)	Power (mW)	$D^2 \cdot P$ $(s^2 \cdot W)$
0.9	272.8	0.765	$4.11 \cdot 10^{-23}$
1.0	207.8	0.852	$3.67 \cdot 10^{-23}$
1.1	201.3	0.947	$3.83 \cdot 10^{-23}$

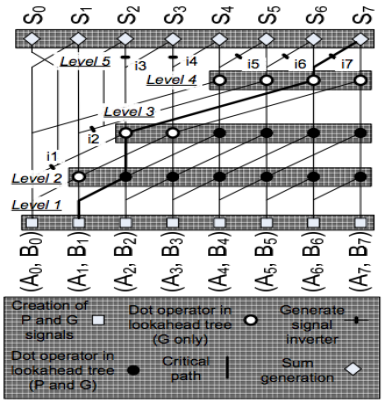

Figure 2: Kogge-Stone Tree Graph Topology [1]

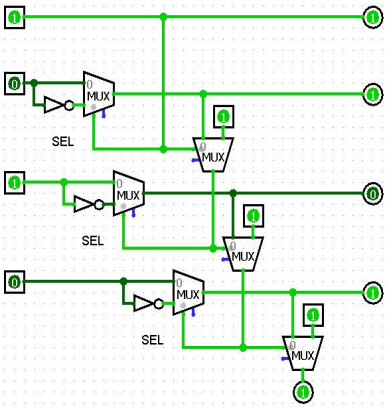

Figure 3: 4-bit 2COMP Topology (*Logisim*)

the greatest delay, which proved greater than the others. As a result, the adder would offer a large delay to the system. To greatly reduce this delay, the design was changed to a Kogge-Stone Adder, one of the fastest adders available. The delay of the Kogge-Stone Adder was 30% of the mirror adder. The delay was greatly reduce as it parallelized a lot of processes, allowing it to get the final carrier bit in 5 stages (worst case) as it pulls the values from the propagation of the first pair of bits. Figure 2 shows our design's Kogge-Stone Adder tree graph topology.

2.3 Multiplexed 2COMP

The simulation results show that the second greatest delay came from the 2COMP operation. To improve upon this, we changed the design to require fewer components to propagate the signals through. The original design inverted all the input bits, then passed that signal through to an adder, and added it by 1. Given that this design required an adder, the overall function's speed fell. The new design only requires 13 multiplexers for an 8-bit input. This greatly minimized the 2COMP's area, power, and delay. Figure 3 illustrates the minimized area for a 4-bit 2COMP topology.

2.4 Parallel Comparator

Team Awesome's DSP unique arbitrary function is a comparator. Given that there are 16 bits for the input, comparing each bit individually would be very expensive. To improve upon this design, the component utilizes parallel comparison. As a result, this comparator has 4 parts that each have 4 input bits and 2 outputs. The "greater than" and "equal to" outputs are then propagated to each successive block starting from the most significant bit. From this design, the comparator will have a much faster performance. Figure 4 shows our 2-bit Comparator block design.

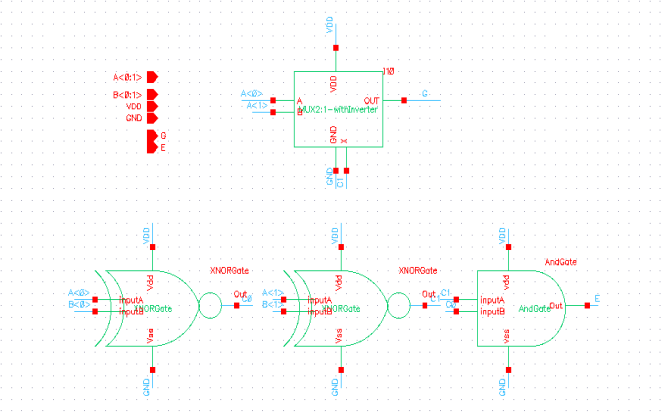

Figure 4: 2-bit Comparator Block Schematic (*Cadence*)

2.5 Register Design

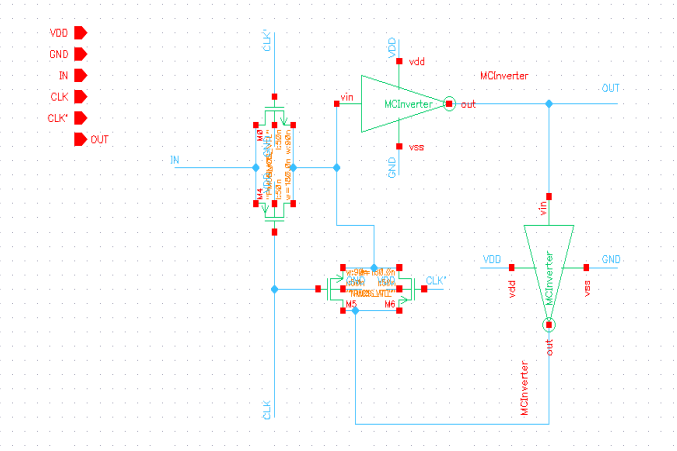

Figure 5: D-Latch Schematic (*Cadence*)

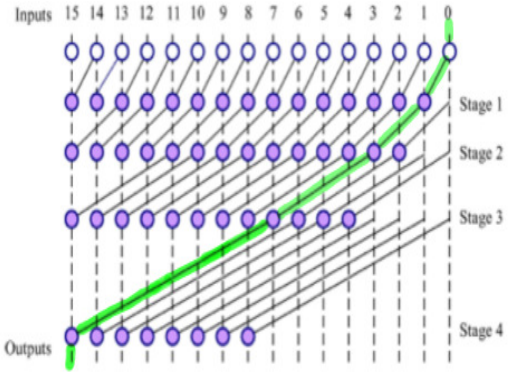

Figure 6: Critical Path of 16-bit Kogge-Stone Adder

To implement the 8-bit register, two D-latches were placed in series to form a master-slave positive-edge-triggered D flip-flop, or register. Figure 5 shows the D-latch design.

3. INNOVATION

3.1 Kogge-Stone Adder

Kogge-Stone parallel prefix adders (PPA) trade power consumption for performance. They are arguably the fastest industrial standard adder. The Kogge-Stone decreases the delay of a mirror adder by 3 times; however it will use more power and cover a greater area. Given that IBL values delay over both power consumption and size (delay $\times$ power^2 $\times$ *area*), speed was our highest priority. Table 3 shows the difference in delays for the two designs.

Figure 6 shows the critical path of the Kogge-Stone Adder.

3.2 Multiplexed 2COMP

Table 3: Adder Delay Comparison

Component	Delay (ps)
Mirror Adder	265.1
Kogge-Stone Adder	127.5

Table 4: Multiplexed 2COMP Delay

ALU Function	Delay (ps)
2COMP	112.2

Table 5: Parallel Comparator Delay

ALU Function	Delay (ps)
2COMP	54.15

The 2COMP was also built to minimize delay. Instead of inverting all the bits and adding 1, we utilized multiplexers. By not utilizing an adder, this component is both much faster and much smaller compared to the naïve design. Table 4 shows the delay of the 2COMP component.

3.3 Parallel Comparator

The comparator speed increases by using four 2-bit blocks as opposed to eight 1-bit comparator blocks. While this may consume slightly more power and be larger, its delay is decreased. Table 5 shows the delay of the Comparator.

3.4 Extensions

Although we believe based on our results that our DSP is more than qualified for IBL's contract, we have more innovative ideas for a next generation IBL DSP. Various options were researched that we began implementing; however, after implementing and simulating, it became apparent that these options proved ineffective given our overall design at this time. The following describe our researched ideas and interests:

3.4.1 Multiplexing V_{DD}

Multiplexing the power supply V_{DD} to power-expensive components appears to be a great way to reduce power loss because it effectively rations the power to only the component in use. Our simulations showed potential with every component except for the adder. Multiplexing the power supply reduces the necessary power for the Kogge-Stone adder to work. To fully take advantage of V_{DD}, the size of the PMOS transistors need to be big enough to drive the power; however, this also increases area as the width is increased.

3.4.2 Encoding Inputs

Encoding the inputs can lead to greater performance with smaller area and power consumption. Encoded inputs would require smaller gates and components, thereby reducing delay, power, and area all together. There would also be decoding that occurs after the operation. With this encoding, the DSP's efficiency would surely increase.

3.4.3 Synthesized Adder

Our design focused on the Kogge-Stone PPL adder; however, we researched other high caliber adders, such as the Brent-Kung Adder. Although the Kogge-Stone adder maximizes performance through the usage of parallel calculations, it fails to minimize area and power consumption. The Brent-Kung Adder is a high performance adder that minimizes overall area and power consumption with a $O(\lg n)$ performance similar to Kogge-Stone. Synthesizing the two could achieve a fast and lightweight adder design. Furthermore,

Table 6: DSP Metrics

Component	Power (μW)	Delay (ps)	Area (μm)
DSP	852	207.77	152.617

Table 7: Delay Breakdown

ALU Function	Delay (ps)
ADD	127.5 + 38.59 = 166.09
2COMP	112.2 + 38.59 = 150.79
SHIFT	38.61 + 38.59 = 77.22
AND	24.4 + 38.59 = 62.99
OR	19.79 + 38.59 = 58.38
NOP/Pass A	38.59
Comparator	54.15 + 38.59 = 89.74

further synthesizing the former with a Han-Carlson Adder (utilized in Intel Pentium 4 microprocessors [2]) would give IBL an adder beyond comparison.

3.4.4 Optimized 2COMP

Although the 2COMP component already possesses innovations in design through the usage of a multiplexing system, further optimization is possible. Specifically, if the 2COMP utilized a parallel network rather than a single series connection, the performance would greatly increase.

4. RESULTS

4.1 Metric

The metric for our DSP design is the following equation:

$$\text{Metric} = \text{Area} \times \text{Delay}^2 \times \text{Power} \tag{1}$$

where Delay is determined by,

$$t_{min} = t_{setup,register} + t_{clk-Q,register} + t_{delay,ALU} \tag{2}$$

Where $t_{setup,register} = 36.5$ ps, $t_{clk-Q,register} = 43.77$ ps, and $t_{delay,ALU} = 127.5$ ps. Therefore,

$$t_{min} = 36.5 \text{ ps } + 43.77 \text{ ps } + 127.5 \text{ ps} \tag{3}$$

$$t_{min} = 207.77 \text{ ps} \tag{4}$$

Given the previous values, duplicated in table 6,

$$\text{Metric} = (152.617\mu m)\,(207.77\text{ps})^2\,(852\mu W) \tag{5}$$

$$\text{Metric} = 5.613 \cdot 10^{-27} m \cdot s^2 \cdot W \tag{6}$$

4.2 Power and Delay Breakdown

Table 7 has the delay breakdown for each component in the DSP. The 38.59 ps is the delay of the 8-to-1 MUX used to select the ALU operation.

Table 8 has the power breakdown for each component in the DSP.

Table 8: Power Breakdown

ALU Function	Power Ratio (%)
ADD	56
2COMP	9.1
SHIFT	4.2
AND	3.3
OR	5.2
NOP/Pass A	1.1
Comparator	15

5. CONCLUSION

In this paper, Team Awesome presented an optimal digital signal processor to Itty Bitty Logic Co. (IBL) using FreePDK 45 nm technology. Given the design requirements, Team Awesome achieved those requirements and went the extra mile to optimize their design using innovative ideas such as the parallel comparator and a highly advanced yet beautifully simple multiplexing two's complement system. The parallel comparator minimizes delay compared to traditional comparator models through the usage of four 2-bit blocks as opposed to traditional 1-bit blocks. The two's complement system developed relies on a multiplexer system as opposed the traditional adder-two's complement employed by our competitors, thereby decreasing overall area and delay. Both of these new and innovative designs increase the overall performance of the DSP greatly. Furthermore, utilizing well-known high performance topologies such as the Kogge-Stone parallel prefix adder to provide a $O(\lg n)$ performance only add to our DSP's overall performance. This Kogge-Stone adder produces a critical path delay twelve times less than the traditional Ripple-Carry Adder. Lastly, the extensions listed indicate Team Awesome's current research interests and desire to continue building a better, next-generation DSP as well as continue a relationship with IBL. Team Awesome's goal is to ultimately maximize performance while minimizing both area and power consumption, and such research extensions as multiplexing the power supply or encoding the inputs indicate these desires. In conclusion, Team Awesome adheres to each and every design requirement of IBL; moreover, given the extent of Team Awesome's implementation of our innovative designs and ideas, Team Awesome is not the contract IBL deserves, but the contract IBL needs right now.

6. ACKNOWLEDGMENTS

We would like to thank Itty Bitty Logic Co. for giving us the opportunity to compete for their contract. We would also like to thank North Carolina State for giving access to their FreePDK technology in Cadence Spectre and Virtuoso.

7. REFERENCES

[1] Bilicki, Ivan. "8-Bit Carry Lookahead Logarithmic Kogge-Stone Adder in CMOS 0.18μm Technology."
N.p., n.d. Web. 01 May 2013.
<http://ppeev.com/pdf_files/VLSI_LogAdder.pdf>.
[2] "Fast and Area-Efficeint VLSI Adders - MSL." Fast and Area-Efficeint VLSI Adders - MSL.
N.p., n.d. Web. 01 May 2013.
<https://sites.google.com/site/ysmslys/research/fast-and-area-efficeint-vlsi-adders>.

APPENDIX

On the following pages are the various netlists, schematics, and simulations pertaining to our DSP design.

The following 4 figures (7-10) show the simulation results of adder which is the critical delay path of the ALU. The simulation is 2.5 cycle long (at the max frequency). The control vector to the ALU unit is 010, Input B is 00000001, Input A is 1111111X and the last bit of Input A is flipping between 0 and 1 every .3ns. Clock is set at the maximum period. As expected, Carryout is a one and the ALU outputs are all zero when last bit of Input A is 1. Otherwise, Carryout is zero and ALU outputs are all 1 when the last bit of A is zero. Con is the buffer's intermediate value after the first register to the ALU.

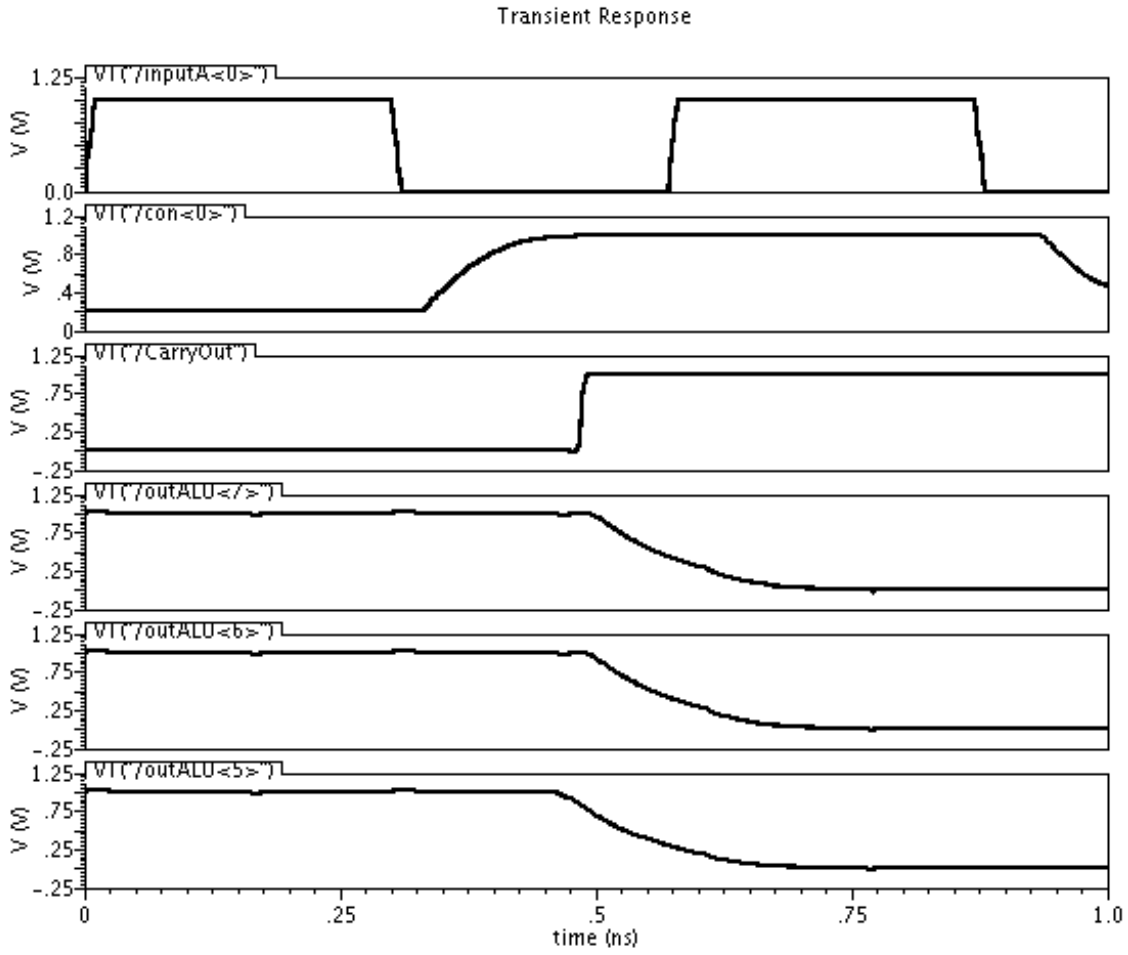

Figure 7: Critical Path Transient Response of ALU

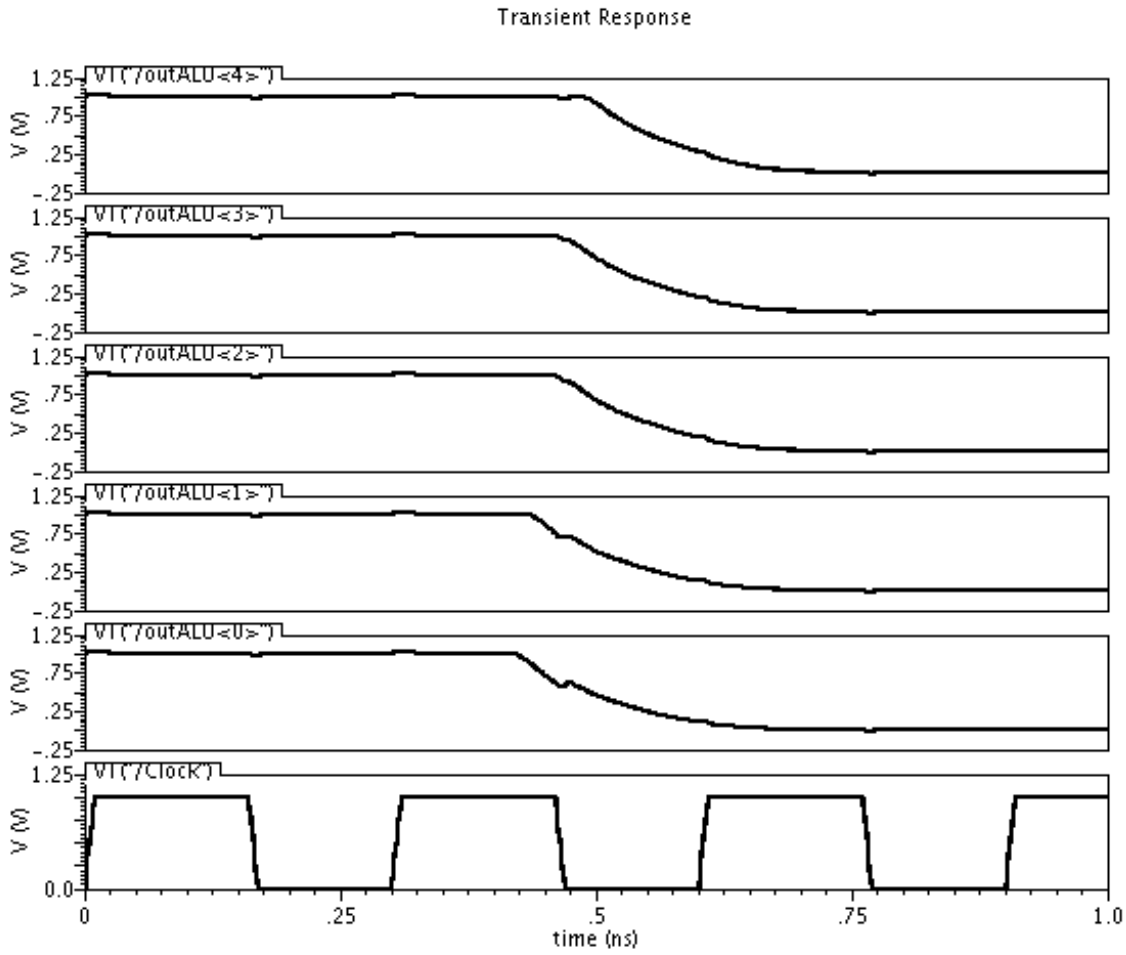

Figure 8: Critical Path Transient Response of ALU

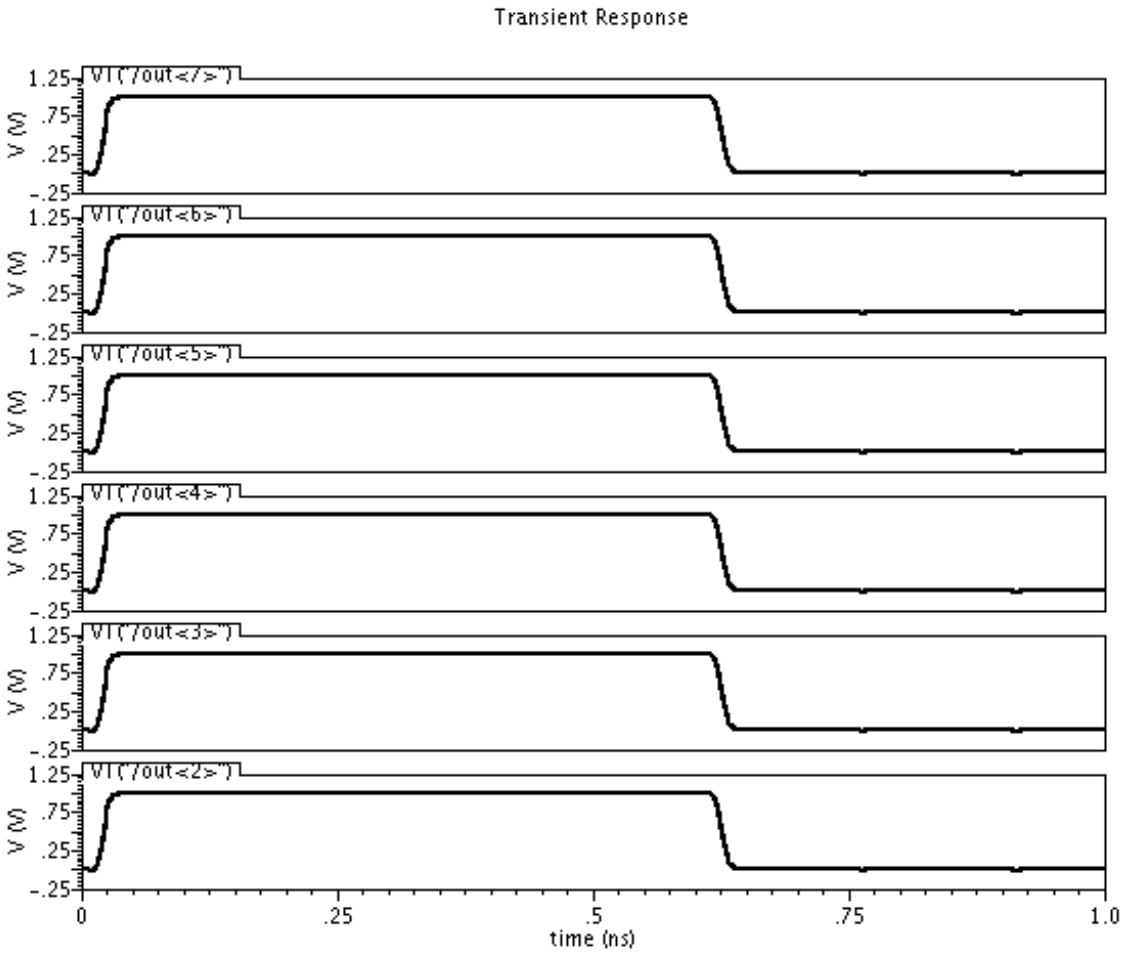

Figure 9: Critical Path Transient Response of ALU

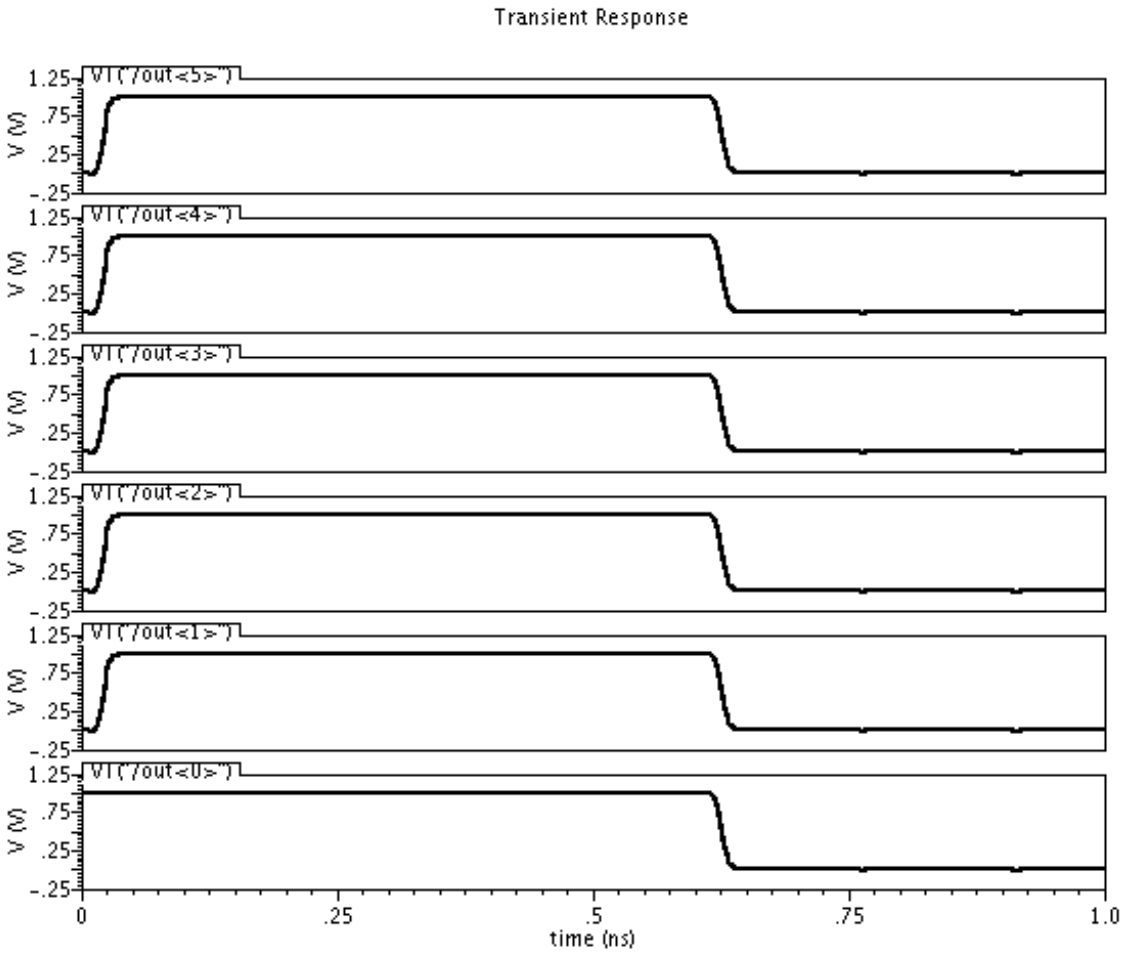

Figure 10: Critical Path Transient Response of ALU

The inputs for the comparator are A and B. The output is rightfully named "output" on the graph. While there are 4 bits, only the least significant bit is used to determine the greater than, equal to function. If it is greater than or equal to, the output is 1. If the value is less than, the output bit will be 0. The select bit for this function is 111. Figures 11-14 show the transient response of the comparator.

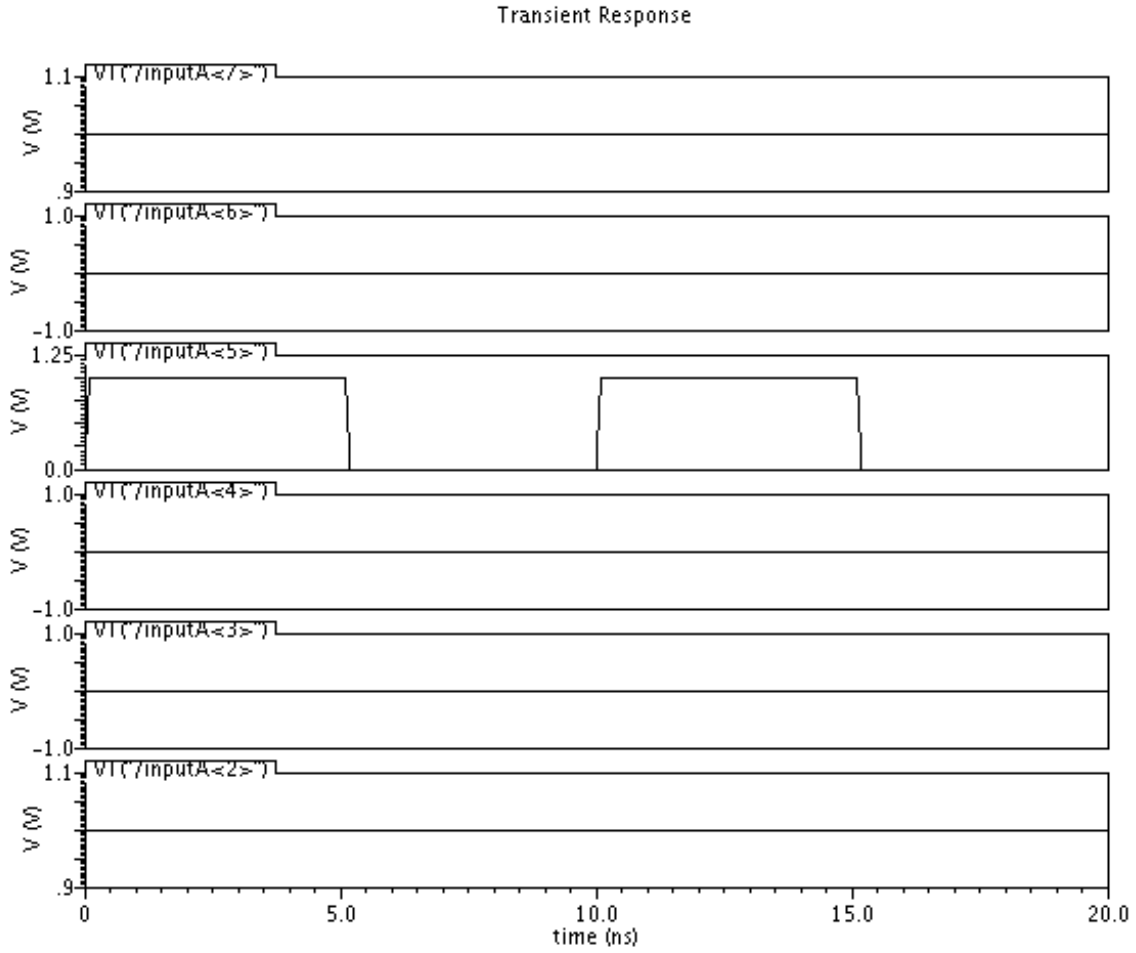

Figure 11: Comparator - InputA¡2:7¿

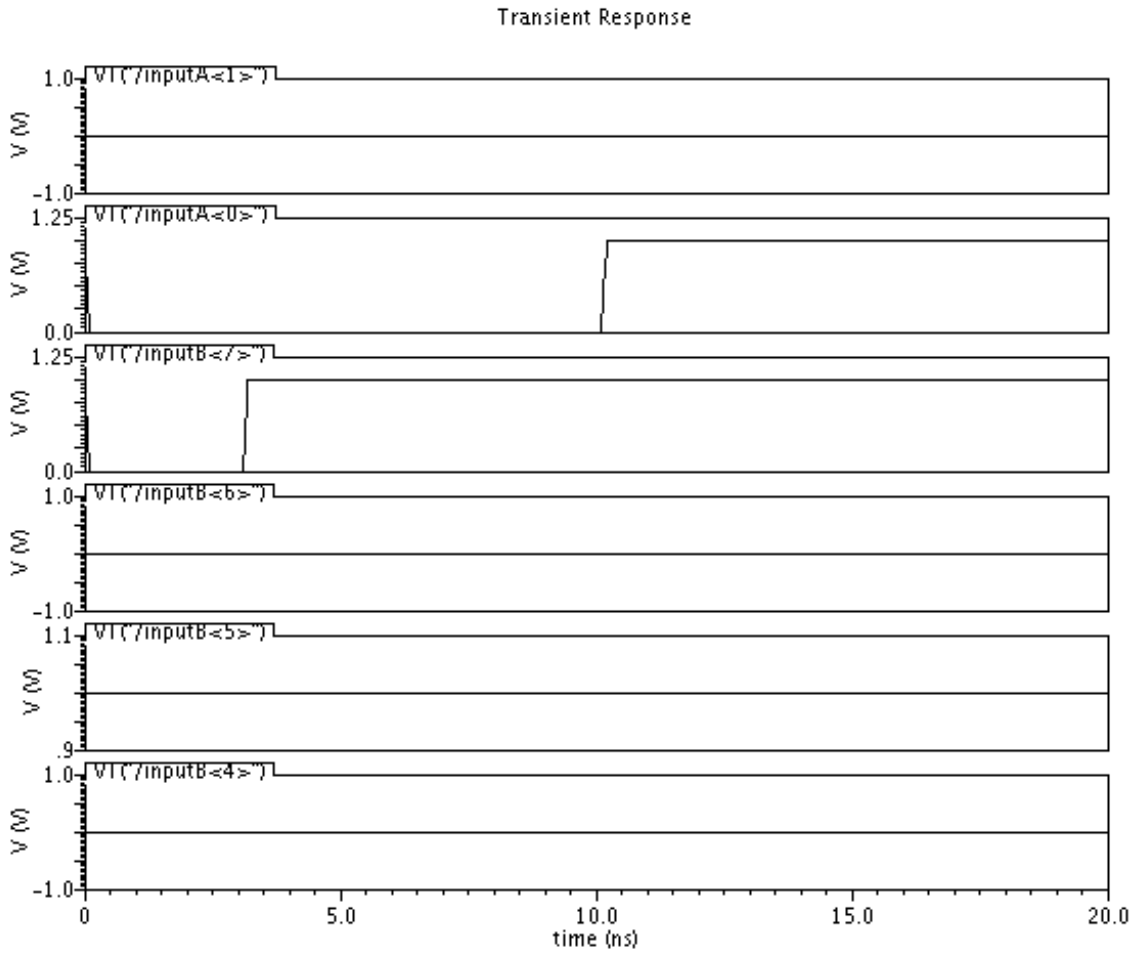

Figure 12: Comparator - InputB¡4:7¿, InputA¡0:1¿

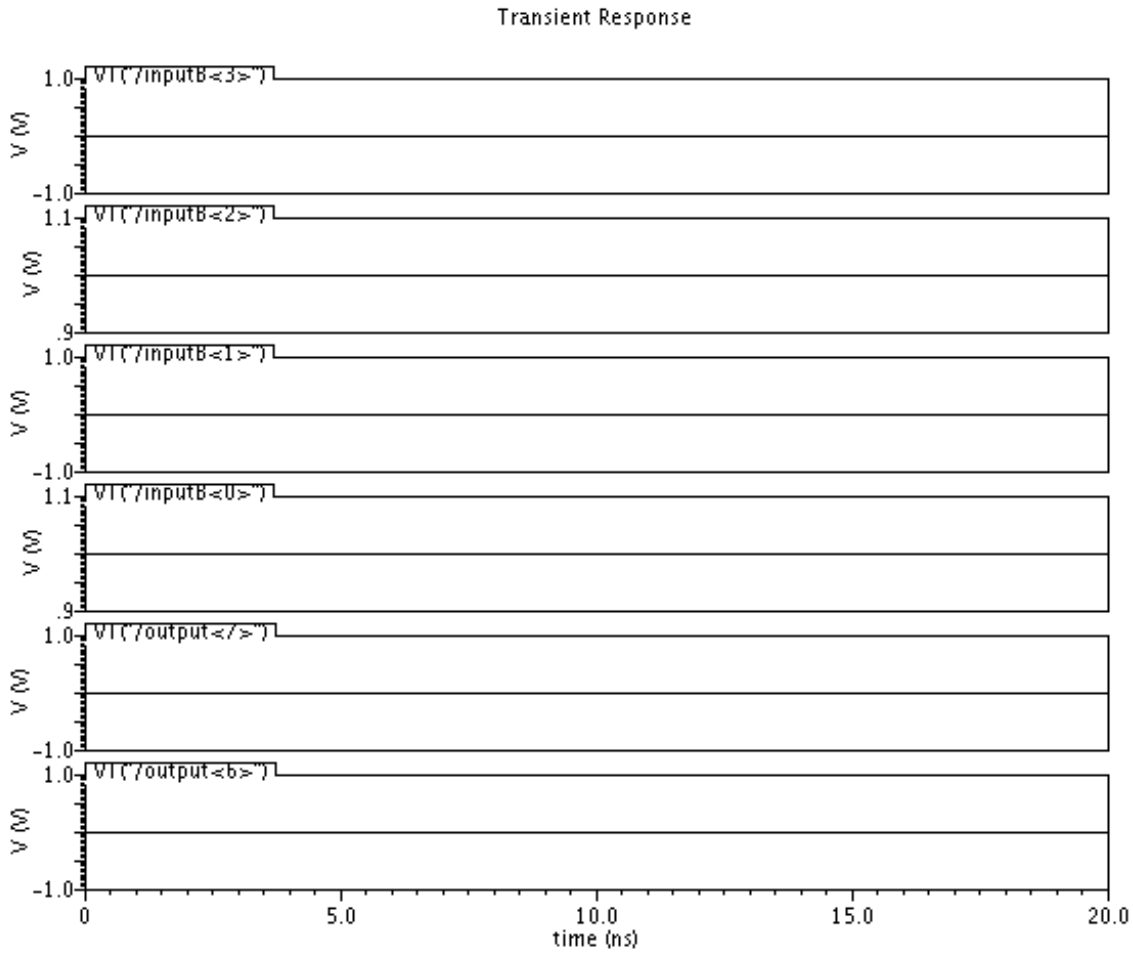

Figure 13: Comparator - Output¡6:7¿, InputB¡0:3¿

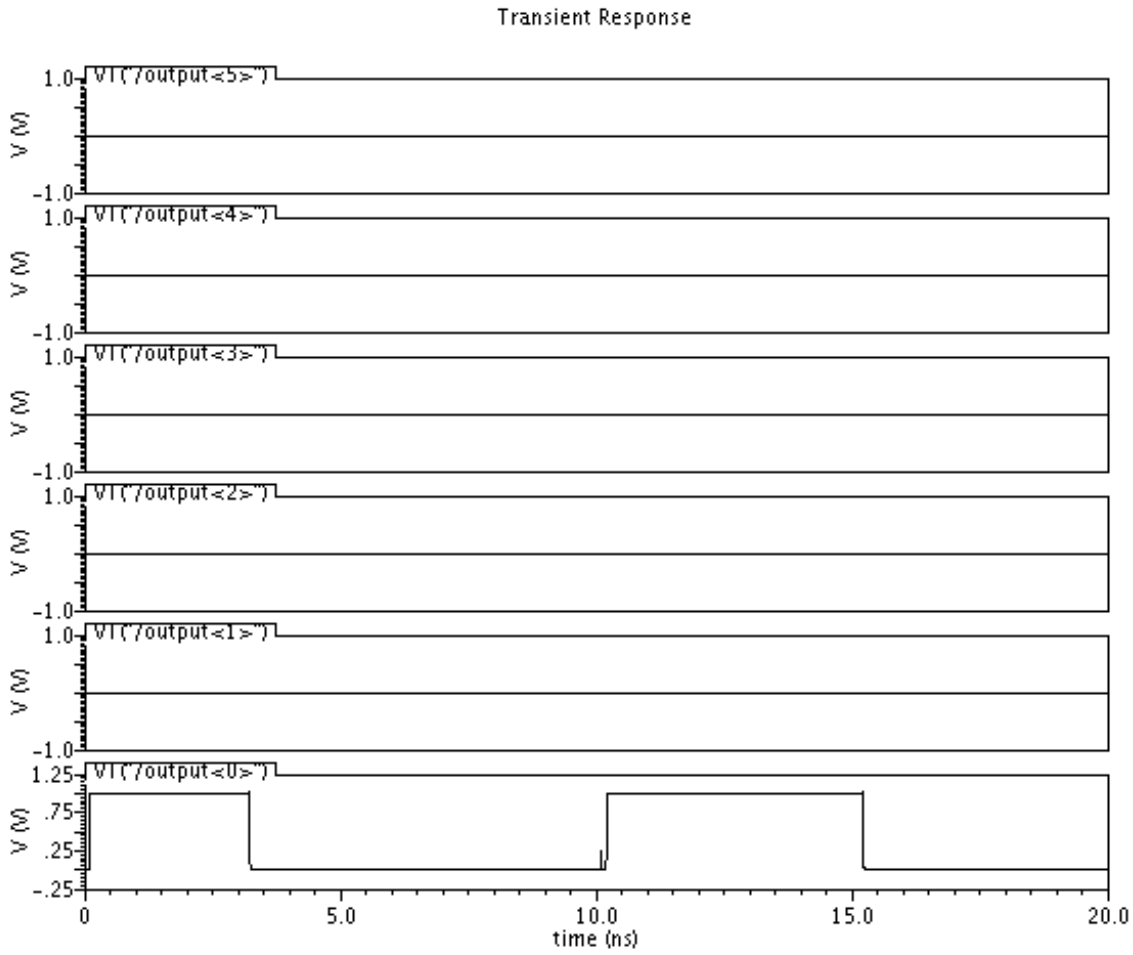

Figure 14: Comparator - Output¡0:5¿

Figures 15 and 16 show the energy simulation of the DSP.

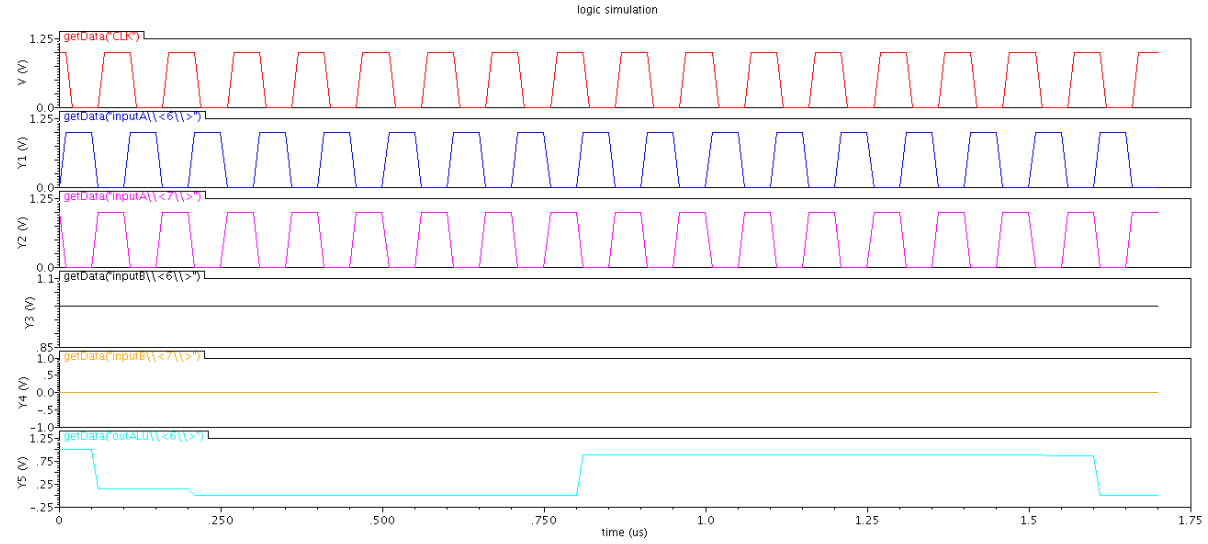

Figure 15: Clock, InputA, InputB, ALU Output (from top down)

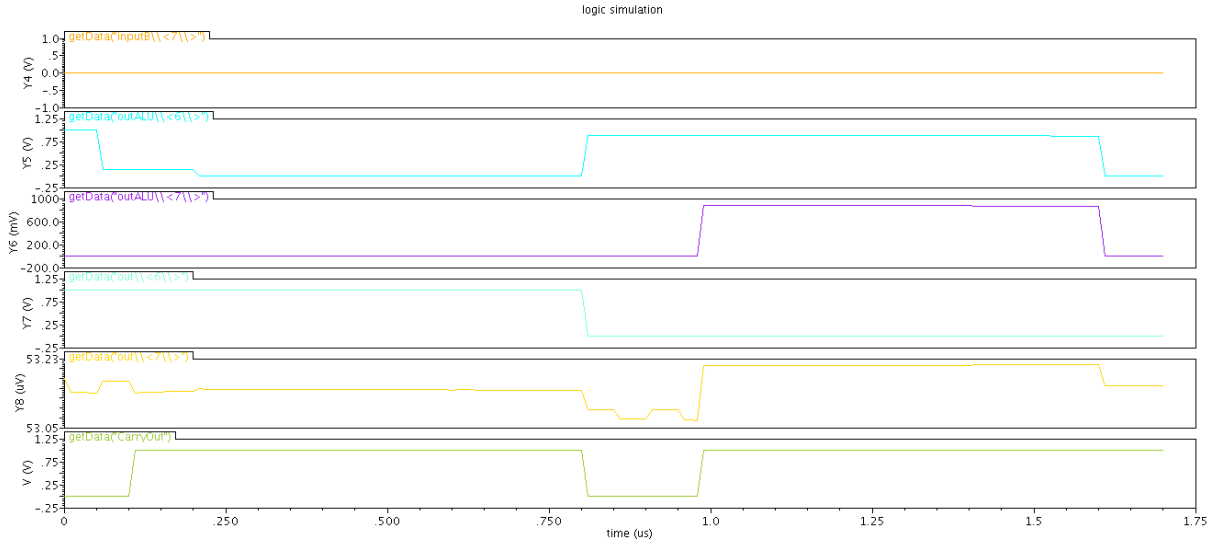

Figure 16: Final Output and carry out

Figure 17 shows the final high-level design of the DSP with the test buffers.

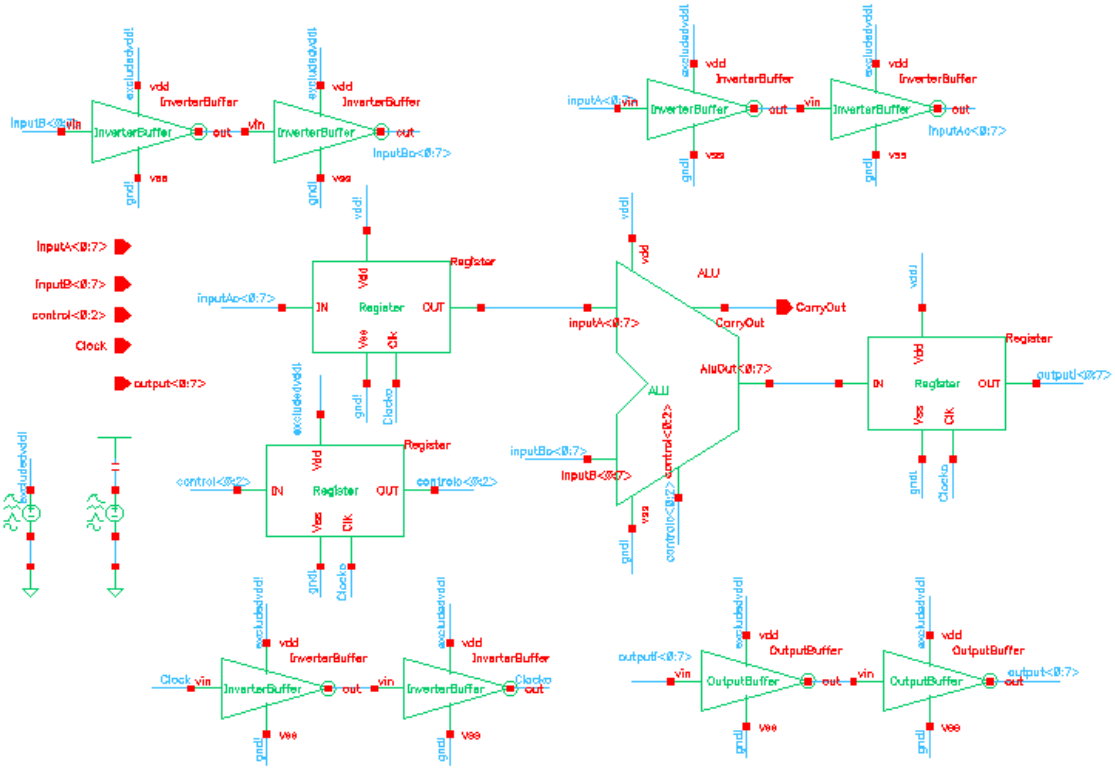

Figure 17: Final Schematic with Buffers included

Figures 18-21 show different components of our Kogge-Stone Adder.

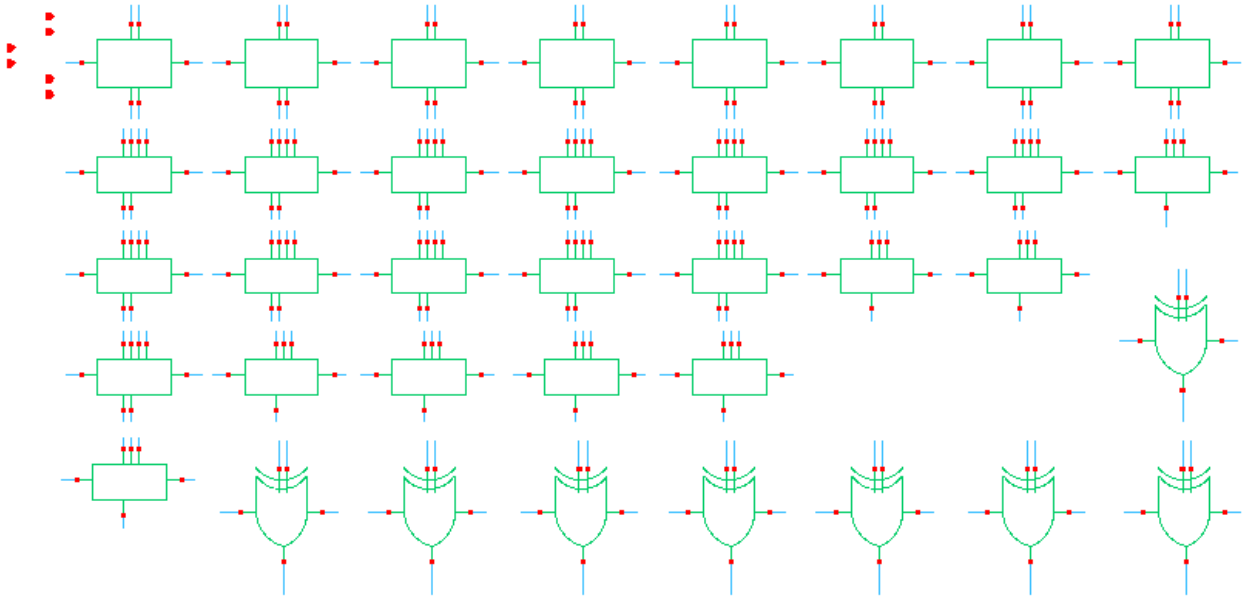

Figure 18: Entire topology of Kogge-Stone Adder

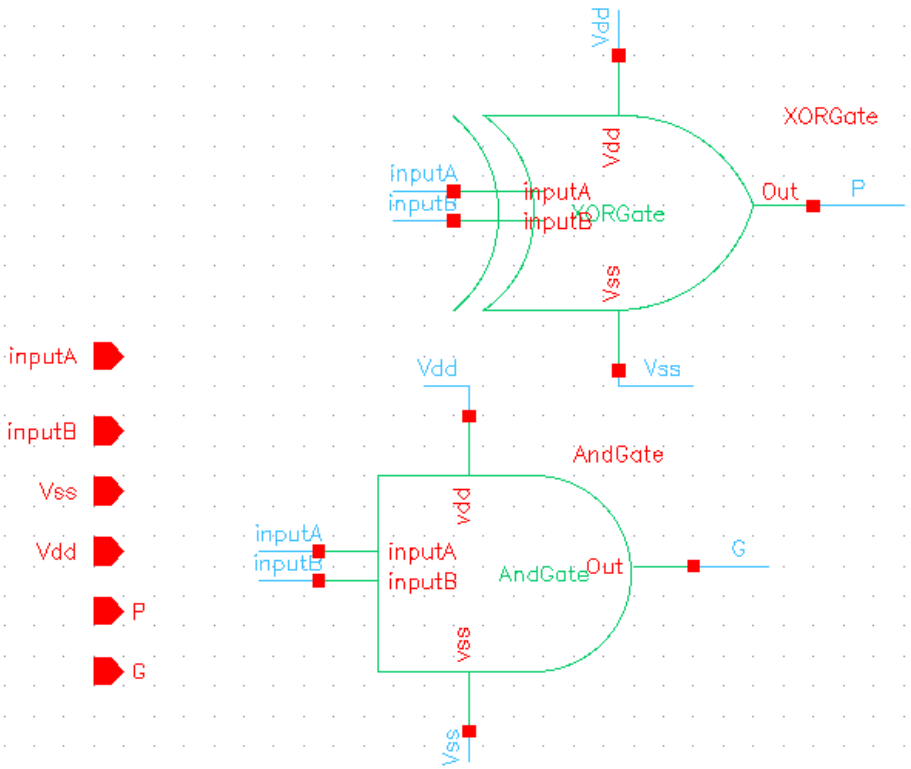

Figure 19: Bit Block of Kogge-Stone Adder

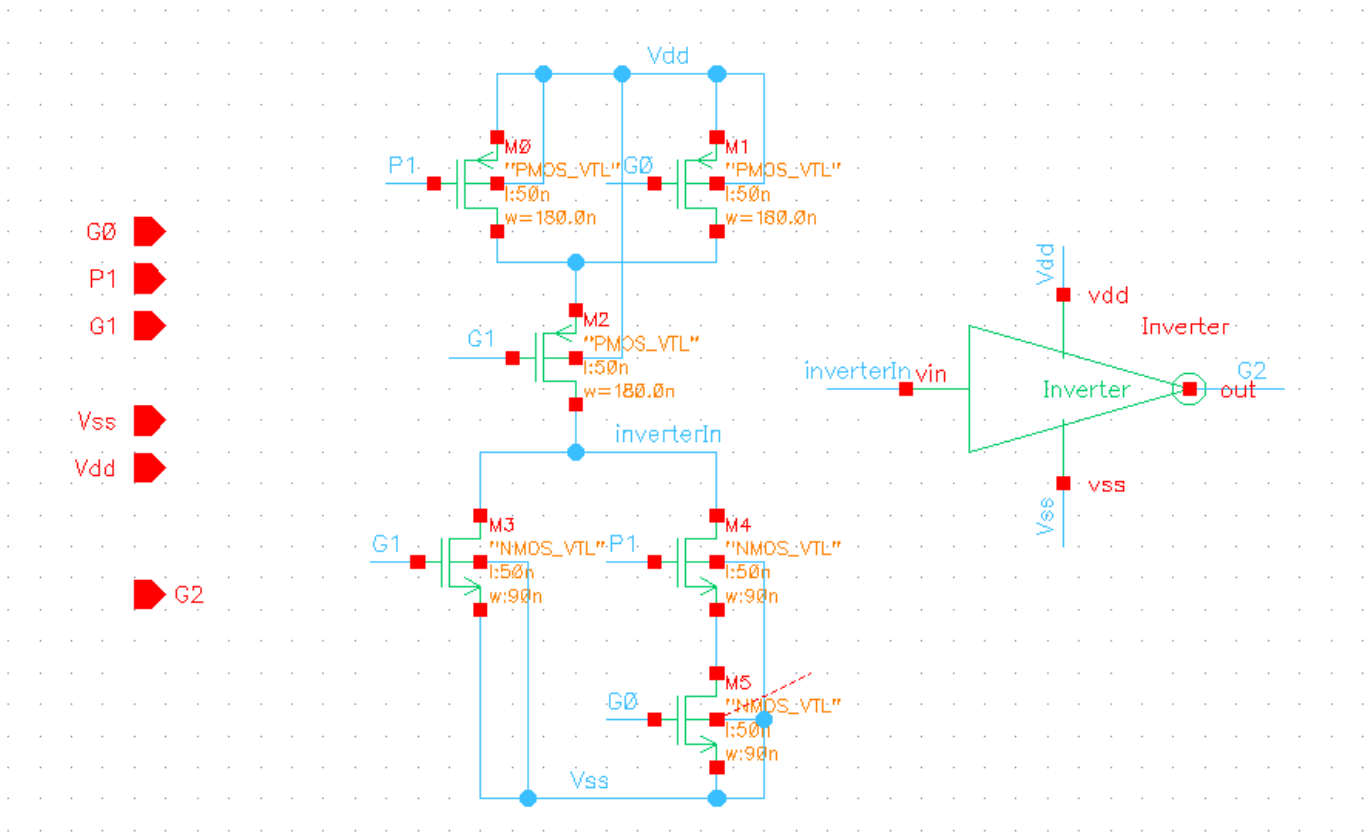

Figure 20: Group of Generate component of Kogge-Stone Adder

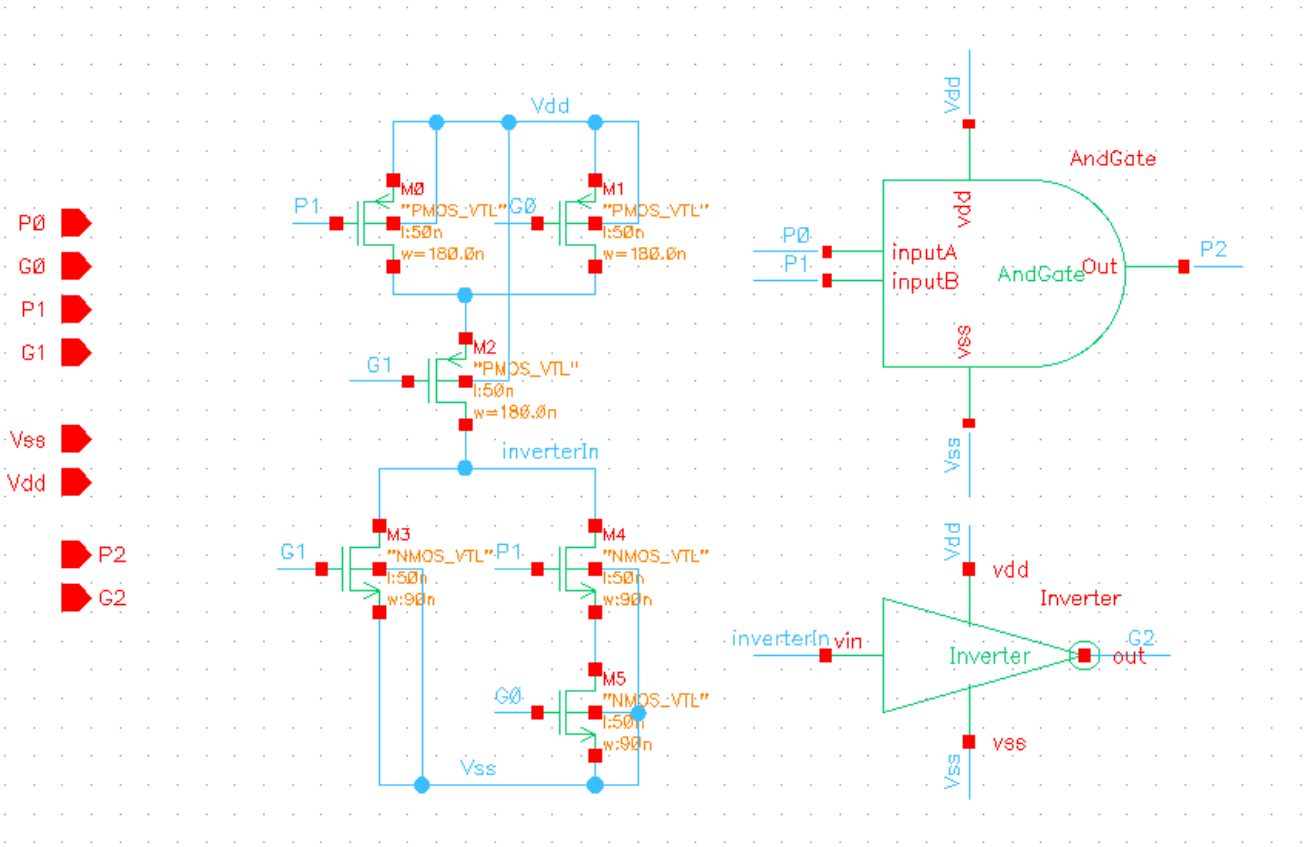

Figure 21: Group of Generate Propagate component of Kogge-Stone Adder

The initial Ocean script used to aid in testing is included:

```
;; DEFINE OUTPUT PRINT FILE
of = outfile( "output.txt" "w" )

;; SET SIMULATOR
simulator( 'spectre )

;; SET DESIGN
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; NOTE: the design name can only be 'netlist'.
;; However, a directory name can be added, like
;;  "./mydir/netlist"
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
design( "netlist" )


;; SET parameters as needed
;; any parameters defined in the netlist will keep those values - do not redefine here
desVar( "pvdd" 1.0 )
desVar( "p" .1n)
pvdd=1.0;

;; SET TEMPERATURE
temp( 25 )

;; SET RESULT DIRECTORY
sprintf(dir "./TranResults")
resultsDir( dir )

 ;; RUN ANALYSIS
 analysis( 'tran ?start 0 ?stop 80n ?step .1n ?strobeperiod .1n ?errpreset 'conservative )
 save( 'all )
 run()

 ;; PLOT THE SIGNALS
 selectResults('tran)
plot(getData("ShifterInput\\<0\\>") getData("ShifterInput\\<1\\>")
     getData("ShifterInput\\<2\\>") getData("ShifterInput\\<3\\>")
     getData("ShifterInput\\<4\\>") getData("ShifterInput\\<5\\>")
     getData("ShifterInput\\<6\\>") getData("ShifterInput\\<7\\>")  )
plot(getData("ShifterOutput\\<0\\>") getData("ShifterOutput\\<1\\>")
     getData("ShifterOutput\\<2\\>") getData("ShifterOutput\\<3\\>")
     getData("ShifterOutput\\<4\\>") getData("ShifterOutput\\<5\\>")
     getData("ShifterOutput\\<6\\>") getData("ShifterOutput\\<7\\>"))


close( of )
```

The final Ocean script used to aid in testing is included:

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; OCEAN example to simulate the
;;  rising and falling times of an inverter
;;  with FreePDK
;;
;; Ben Calhoun
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;


;; DEFINE OUTPUT PRINT FILE
of = outfile( "output.txt" "w" )

;; SET SIMULATOR
simulator( 'spectre )

;; SET DESIGN
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;; NOTE: the design name can only be 'netlist'.
;; However, a directory name can be added, like
;;  "./mydir/netlist"
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
design( "netlist" )


;; SET parameters as needed
;; any parameters defined in the netlist will keep those values - do not redefine here
desVar( "pvdd" 1.0 )
desVar( "pd" 100n)

pd=100n;
;; SET TEMPERATURE
temp( 25 )

;; SET RESULT DIRECTORY
sprintf(dir "./TranResults")
resultsDir( dir )

 ;; RUN ANALYSIS
 analysis( 'tran ?start 0 ?stop 17*pd ?step 10n ?strobeperiod 10n ?errpreset 'conservative )
 save( 'all )
 run()


 ;; PLOT THE SIGNALS
;;finding the average iavg*voltage
selectResults('tran)
iavg=average(i("net37"));
plot(getData("CLK") getData("inputA\\<6\\>") getData("inputA\\<7\\>")
     getData("inputB\\<6\\>") getData("inputB\\<7\\>")
     getData("outALU\\<6\\>") getData("outALU\\<7\\>")
     getData("out\\<6\\>") getData("out\\<7\\>") getData("CarryOut"))

close( of )
```

The netlist of the DSP is included:

```
// Generated for: spectre
// Generated on: May  2 06:34:13 2013
// Design library name: DIC
// Design cell name: DSProcessorWBInput
// Design view name: schematic
simulator lang=spectre
global 0 vdd! excludedvdd!
include "/net/plato.ee.Virginia.EDU/app/lib/freepdk45/trunk/
     ncsu_basekit/models/hspice/tran_models/models_nom/PMOS_VTL.inc"
include "/net/plato.ee.Virginia.EDU/app/lib/freepdk45/trunk/
     ncsu_basekit/models/hspice/tran_models/models_nom/NMOS_VTL.inc"
parameters _gpar0 wsoscaler wiscaler

// Library name: DIC
// Cell name: Inverter
// View name: schematic
subckt Inverter out vdd vin vss
    M0 (out vin vss vss) NMOS_VTL w=90n l=50n as=9.45e-15 ad=9.45e-15 \
        ps=300n pd=300n ld=105n ls=105n m=1
    M1 (out vin vdd vdd) PMOS_VTL w=180.0n l=50n as=1.89e-14 ad=1.89e-14 \
        ps=390.0n pd=390.0n ld=105n ls=105n m=1
ends Inverter
// End of subcircuit definition.

// Library name: DIC
// Cell name: OR
// View name: schematic
subckt OR Out inputA inputB vdd vss
    M1 (net33 inputB net24 vdd) PMOS_VTL w=180.0n l=50n as=1.89e-14 \
        ad=1.89e-14 ps=390.0n pd=390.0n ld=105n ls=105n m=1
    M0 (net24 inputA vdd vdd) PMOS_VTL w=180.0n l=50n as=1.89e-14 \
        ad=1.89e-14 ps=390.0n pd=390.0n ld=105n ls=105n m=1
    M3 (net33 inputB vss vss) NMOS_VTL w=90n l=50n as=9.45e-15 ad=9.45e-15 \
        ps=300n pd=300n ld=105n ls=105n m=1
    M2 (net33 inputA vss vss) NMOS_VTL w=90n l=50n as=9.45e-15 ad=9.45e-15 \
        ps=300n pd=300n ld=105n ls=105n m=1
    I1 (Out vdd net33 vss) Inverter
ends OR
// End of subcircuit definition.

// Library name: DIC
// Cell name: AND
// View name: schematic
subckt AND Out inputA inputB vdd vss
    M3 (net5 inputB vss vss) NMOS_VTL w=90n l=50n as=9.45e-15 ad=9.45e-15 \
        ps=300n pd=300n ld=105n ls=105n m=1
    M2 (net17 inputA net5 vss) NMOS_VTL w=90n l=50n as=9.45e-15 \
        ad=9.45e-15 ps=300n pd=300n ld=105n ls=105n m=1
    M0 (net17 inputA vdd vdd) PMOS_VTL w=180.0n l=50n as=1.89e-14 \
        ad=1.89e-14 ps=390.0n pd=390.0n ld=105n ls=105n m=1
    M1 (net17 inputB vdd vdd) PMOS_VTL w=180.0n l=50n as=1.89e-14 \
        ad=1.89e-14 ps=390.0n pd=390.0n ld=105n ls=105n m=1
    I2 (Out vdd net17 vss) Inverter
```

```
ends AND
// End of subcircuit definition.

// Library name: DIC
// Cell name: MUX2:1-withInverter
// View name: schematic
subckt _sub0 A B GND OUT VDD X
    M2 (OUT net54 B GND) NMOS_VTL w=90n l=50n as=9.45e-15 ad=9.45e-15 \
        ps=300n pd=300n ld=105n ls=105n m=1
    M0 (OUT X A GND) NMOS_VTL w=90n l=50n as=9.45e-15 ad=9.45e-15 ps=300n \
        pd=300n ld=105n ls=105n m=1
    M3 (OUT X B VDD) PMOS_VTL w=180.0n l=50n as=1.89e-14 ad=1.89e-14 \
        ps=390.0n pd=390.0n ld=105n ls=105n m=1
    M1 (OUT net54 A VDD) PMOS_VTL w=180.0n l=50n as=1.89e-14 ad=1.89e-14 \
        ps=390.0n pd=390.0n ld=105n ls=105n m=1
    I0 (net54 VDD X GND) Inverter
ends _sub0
// End of subcircuit definition.

// Library name: DIC
// Cell name: Complement1-new
// View name: schematic
subckt _sub1 CIN COUT GND IN INV OUT VDD
    I1 (VDD OUT GND COUT VDD CIN) _sub0
    I0 (INV IN GND OUT VDD CIN) _sub0
ends _sub1
// End of subcircuit definition.

// Library name: DIC
// Cell name: Complement
// View name: schematic
subckt Complement GND IN\<0\> IN\<1\> IN\<2\> IN\<3\> IN\<4\> IN\<5\> \
        IN\<6\> IN\<7\> INV\<1\> INV\<2\> INV\<3\> INV\<4\> INV\<5\> \
        INV\<6\> INV\<7\> OUT\<0\> OUT\<1\> OUT\<2\> OUT\<3\> OUT\<4\> \
        OUT\<5\> OUT\<6\> OUT\<7\> VDD
    I5 (VDD IN\<0\> GND C1 VDD IN\<0\>) _sub0
    I19 (C1 C2 GND IN\<1\> INV\<1\> OUT\<1\> VDD) _sub1
    I21 (C3 C4 GND IN\<3\> INV\<3\> OUT\<3\> VDD) _sub1
    I20 (C2 C3 GND IN\<2\> INV\<2\> OUT\<2\> VDD) _sub1
    I22 (C4 C5 GND IN\<4\> INV\<4\> OUT\<4\> VDD) _sub1
    I23 (C5 C6 GND IN\<5\> INV\<5\> OUT\<5\> VDD) _sub1
    I25 (C7 net104 GND IN\<7\> INV\<7\> OUT\<7\> VDD) _sub1
    I24 (C6 C7 GND IN\<6\> INV\<6\> OUT\<6\> VDD) _sub1
    I17 (IN\<0\> OUT\<0\>) iprobe
ends Complement
// End of subcircuit definition.

// Library name: DIC
// Cell name: MUX2:1
// View name: schematic
subckt _sub2 A B GND OUT VDD X X\#
    M2 (OUT X\# B GND) NMOS_VTL w=90n l=50n as=9.45e-15 ad=9.45e-15 \
        ps=300n pd=300n ld=105n ls=105n m=1
    M0 (OUT X A GND) NMOS_VTL w=90n l=50n as=9.45e-15 ad=9.45e-15 ps=300n \
```

```
        pd=300n ld=105n ls=105n m=1
    M3 (OUT X B VDD) PMOS_VTL w=180.0n l=50n as=1.89e-14 ad=1.89e-14 \
        ps=390.0n pd=390.0n ld=105n ls=105n m=1
    M1 (OUT X\# A VDD) PMOS_VTL w=180.0n l=50n as=1.89e-14 ad=1.89e-14 \
        ps=390.0n pd=390.0n ld=105n ls=105n m=1
ends _sub2
// End of subcircuit definition.

// Library name: DIC
// Cell name: MUX8:1
// View name: schematic
subckt _sub3 C0 C0\# C1 C1\# C2 C2\# D0 D1 D2 D3 D4 D5 D6 D7 GND OUT VDD
    I7 (D6 D7 GND net38 VDD C0\# C0) _sub2
    I9 (D2 D3 GND net52 VDD C0\# C0) _sub2
    I8 (D4 D5 GND net45 VDD C0\# C0) _sub2
    I10 (D0 D1 GND net59 VDD C0\# C0) _sub2
    I11 (net59 net52 GND net31 VDD C1\# C1) _sub2
    I12 (net45 net38 GND net24 VDD C1\# C1) _sub2
    I13 (net31 net24 GND OUT VDD C2\# C2) _sub2
ends _sub3
// End of subcircuit definition.

// Library name: DIC
// Cell name: Shifter
// View name: schematic
subckt Shifter c0 c0\# c1 c1\# c2 c2\# input\<0\> input\<1\> input\<2\> \
        input\<3\> input\<4\> input\<5\> input\<6\> input\<7\> out\<0\> \
        out\<1\> out\<2\> out\<3\> out\<4\> out\<5\> out\<6\> out\<7\> vdd \
        vss zeropin\<0\> zeropin\<1\> zeropin\<2\> zeropin\<3\>
    MUX8\:1\<0\> (c0 c0\# c1 c1\# c2 c2\# input\<1\> input\<2\> input\<3\> \
        input\<4\> zeropin\<0\> zeropin\<0\> zeropin\<0\> zeropin\<0\> vss \
        out\<0\> vdd) _sub3
    MUX8\:1\<1\> (c0 c0\# c1 c1\# c2 c2\# input\<2\> input\<3\> input\<4\> \
        input\<5\> input\<0\> zeropin\<1\> zeropin\<1\> zeropin\<1\> vss \
        out\<1\> vdd) _sub3
    MUX8\:1\<2\> (c0 c0\# c1 c1\# c2 c2\# input\<3\> input\<4\> input\<5\> \
        input\<6\> input\<1\> input\<0\> zeropin\<2\> zeropin\<2\> vss \
        out\<2\> vdd) _sub3
    MUX8\:1\<3\> (c0 c0\# c1 c1\# c2 c2\# input\<4\> input\<5\> input\<6\> \
        input\<7\> input\<2\> input\<1\> input\<0\> zeropin\<3\> vss \
        out\<3\> vdd) _sub3
    MUX8\:1\<4\> (c0 c0\# c1 c1\# c2 c2\# input\<5\> input\<6\> input\<7\> \
        zeropin\<0\> input\<3\> input\<2\> input\<1\> input\<0\> vss \
        out\<4\> vdd) _sub3
    MUX8\:1\<5\> (c0 c0\# c1 c1\# c2 c2\# input\<6\> input\<7\> \
        zeropin\<0\> zeropin\<1\> input\<4\> input\<3\> input\<2\> \
        input\<1\> vss out\<5\> vdd) _sub3
    MUX8\:1\<6\> (c0 c0\# c1 c1\# c2 c2\# input\<7\> zeropin\<0\> \
        zeropin\<1\> zeropin\<2\> input\<5\> input\<4\> input\<3\> \
        input\<2\> vss out\<6\> vdd) _sub3
    MUX8\:1\<7\> (c0 c0\# c1 c1\# c2 c2\# 0 zeropin\<1\> zeropin\<2\> \
        zeropin\<3\> input\<6\> input\<5\> input\<4\> input\<3\> vss \
        out\<7\> vdd) _sub3
ends Shifter
```

```
// End of subcircuit definition.

// Library name: DIC
// Cell name: XNOR
// View name: schematic
subckt XNOR Out Vdd Vss inputA inputB
    M0 (connect inputAb inputB Vss) NMOS_VTL w=90n l=50n as=9.45e-15 \
        ad=9.45e-15 ps=300n pd=300n ld=105n ls=105n m=1
    I7 (Out Vdd connect Vss) Inverter
    I5 (inputAb Vdd inputA Vss) Inverter
    I4 (connect inputA inputB inputAb) Inverter
    M1 (connect inputA inputB Vdd) PMOS_VTL w=180.0n l=50n as=1.89e-14 \
        ad=1.89e-14 ps=390.0n pd=390.0n ld=105n ls=105n m=1
ends XNOR
// End of subcircuit definition.

// Library name: DIC
// Cell name: Comparator2bit
// View name: schematic
subckt Comparator2bit A\<0\> A\<1\> B\<0\> B\<1\> E G GND VDD
    I7 (C0 VDD GND A\<0\> B\<0\>) XNOR
    I8 (C1 VDD GND A\<1\> B\<1\>) XNOR
    I3 (E C1 C0 VDD GND) AND
    I10 (A\<0\> A\<1\> GND G VDD C1) _sub0
ends Comparator2bit
// End of subcircuit definition.

// Library name: DIC
// Cell name: Comparator
// View name: schematic
subckt Comparator A\<0\> A\<1\> A\<2\> A\<3\> A\<4\> A\<5\> A\<6\> A\<7\> \
        B\<0\> B\<1\> B\<2\> B\<3\> B\<4\> B\<5\> B\<6\> B\<7\> GND \
        OUT\<0\> OUT\<1\> OUT\<2\> OUT\<3\> OUT\<4\> OUT\<5\> OUT\<6\> \
        OUT\<7\> VDD
    I3 (A\<0\> A\<1\> B\<0\> B\<1\> C\<0\> D\<0\> GND VDD) Comparator2bit
    I2 (A\<2\> A\<3\> B\<2\> B\<3\> C\<1\> D\<1\> GND VDD) Comparator2bit
    I1 (A\<4\> A\<5\> B\<4\> B\<5\> C\<2\> D\<2\> GND VDD) Comparator2bit
    I0 (A\<6\> A\<7\> B\<6\> B\<7\> C\<3\> D\<3\> GND VDD) Comparator2bit
    I9 (D\<2\> D\<3\> GND F\<1\> VDD C\<3\>) _sub0
    I8 (D\<0\> D\<1\> GND F\<0\> VDD C\<1\>) _sub0
    I10 (F\<0\> F\<1\> GND H VDD E\<1\>) _sub0
    I13 (I E\<1\> E\<0\> VDD GND) AND
    I12 (E\<0\> C\<1\> C\<0\> VDD GND) AND
    I11 (E\<1\> C\<3\> C\<2\> VDD GND) AND
    I14 (OUT\<0\> H I VDD GND) OR
    I15\<0\> (GND OUT\<1\>) iprobe
    I15\<1\> (GND OUT\<2\>) iprobe
    I15\<2\> (GND OUT\<3\>) iprobe
    I15\<3\> (GND OUT\<4\>) iprobe
    I15\<4\> (GND OUT\<5\>) iprobe
    I15\<5\> (GND OUT\<6\>) iprobe
    I15\<6\> (GND OUT\<7\>) iprobe
ends Comparator
// End of subcircuit definition.
```

```
// Library name: DIC
// Cell name: XOR
// View name: schematic
subckt XOR Out Vdd Vss inputA inputB
    M0 (connect inputAb inputB Vss) NMOS_VTL w=90n l=50n as=9.45e-15 \
        ad=9.45e-15 ps=300n pd=300n ld=105n ls=105n m=1
    I8 (Out Vdd net20 Vss) Inverter
    I7 (net20 Vdd connect Vss) Inverter
    I5 (inputAb Vdd inputA Vss) Inverter
    I4 (connect inputA inputB inputAb) Inverter
    M1 (connect inputA inputB Vdd) PMOS_VTL w=180.0n l=50n as=1.89e-14 \
        ad=1.89e-14 ps=390.0n pd=390.0n ld=105n ls=105n m=1
ends XOR
// End of subcircuit definition.

// Library name: DIC
// Cell name: KSAdder_BitBlock
// View name: schematic
subckt KSAdder_BitBlock G P Vdd Vss inputA inputB
    I4 (G inputA inputB Vdd Vss) AND
    I3 (P Vdd Vss inputA inputB) XOR
ends KSAdder_BitBlock
// End of subcircuit definition.

// Library name: DIC
// Cell name: KSAdder_Group_PG
// View name: schematic
subckt KSAdder_Group_PG G0 G1 G2 P0 P1 P2 Vdd Vss
    I0 (P2 P0 P1 Vdd Vss) AND
    M2 (inverterIn G1 net17 Vdd) PMOS_VTL w=180.0n l=50n as=1.89e-14 \
        ad=1.89e-14 ps=390.0n pd=390.0n ld=105n ls=105n m=1
    M1 (net17 G0 Vdd Vdd) PMOS_VTL w=180.0n l=50n as=1.89e-14 ad=1.89e-14 \
        ps=390.0n pd=390.0n ld=105n ls=105n m=1
    M0 (net17 P1 Vdd Vdd) PMOS_VTL w=180.0n l=50n as=1.89e-14 ad=1.89e-14 \
        ps=390.0n pd=390.0n ld=105n ls=105n m=1
    M5 (net33 G0 Vss Vss) NMOS_VTL w=90n l=50n as=9.45e-15 ad=9.45e-15 \
        ps=300n pd=300n ld=105n ls=105n m=1
    M4 (inverterIn P1 net33 Vss) NMOS_VTL w=90n l=50n as=9.45e-15 \
        ad=9.45e-15 ps=300n pd=300n ld=105n ls=105n m=1
    M3 (inverterIn G1 Vss Vss) NMOS_VTL w=90n l=50n as=9.45e-15 \
        ad=9.45e-15 ps=300n pd=300n ld=105n ls=105n m=1
    I1 (G2 Vdd inverterIn Vss) Inverter
ends KSAdder_Group_PG
// End of subcircuit definition.

// Library name: DIC
// Cell name: KSAdder_Group_G
// View name: schematic
subckt KSAdder_Group_G G0 G1 G2 P1 Vdd Vss
    M2 (inverterIn G1 net17 Vdd) PMOS_VTL w=180.0n l=50n as=1.89e-14 \
        ad=1.89e-14 ps=390.0n pd=390.0n ld=105n ls=105n m=1
    M1 (net17 G0 Vdd Vdd) PMOS_VTL w=180.0n l=50n as=1.89e-14 ad=1.89e-14 \
        ps=390.0n pd=390.0n ld=105n ls=105n m=1
```

```
    M0 (net17 P1 Vdd Vdd) PMOS_VTL w=180.0n l=50n as=1.89e-14 ad=1.89e-14 \
        ps=390.0n pd=390.0n ld=105n ls=105n m=1
    M5 (net33 G0 Vss Vss) NMOS_VTL w=90n l=50n as=9.45e-15 ad=9.45e-15 \
        ps=300n pd=300n ld=105n ls=105n m=1
    M4 (inverterIn P1 net33 Vss) NMOS_VTL w=90n l=50n as=9.45e-15 \
        ad=9.45e-15 ps=300n pd=300n ld=105n ls=105n m=1
    M3 (inverterIn G1 Vss Vss) NMOS_VTL w=90n l=50n as=9.45e-15 \
        ad=9.45e-15 ps=300n pd=300n ld=105n ls=105n m=1
    I1 (G2 Vdd inverterIn Vss) Inverter
ends KSAdder_Group_G
// End of subcircuit definition.

// Library name: DIC
// Cell name: KoggeStoneAdder
// View name: schematic
subckt KoggeStoneAdder CarryOut Vdd Vss inputA\<0\> inputA\<1\> \
        inputA\<2\> inputA\<3\> inputA\<4\> inputA\<5\> inputA\<6\> \
        inputA\<7\> inputB\<0\> inputB\<1\> inputB\<2\> inputB\<3\> \
        inputB\<4\> inputB\<5\> inputB\<6\> inputB\<7\> output\<0\> \
        output\<1\> output\<2\> output\<3\> output\<4\> output\<5\> \
        output\<6\> output\<7\>
    I8 (G2 P2 Vdd Vss inputA\<2\> inputB\<2\>) KSAdder_BitBlock
    I7 (G1 P1 Vdd Vss inputA\<1\> inputB\<1\>) KSAdder_BitBlock
    I6 (G0 P0 Vdd Vss inputA\<0\> inputB\<0\>) KSAdder_BitBlock
    I5 (G3 P3 Vdd Vss inputA\<3\> inputB\<3\>) KSAdder_BitBlock
    I4 (G4 P4 Vdd Vss inputA\<4\> inputB\<4\>) KSAdder_BitBlock
    I3 (G5 P5 Vdd Vss inputA\<5\> inputB\<5\>) KSAdder_BitBlock
    I2 (G6 P6 Vdd Vss inputA\<6\> inputB\<6\>) KSAdder_BitBlock
    I0 (G7 P7 Vdd Vss inputA\<7\> inputB\<7\>) KSAdder_BitBlock
    I22 (TG3 TG7 UG7 TP3 TP7 UP7 Vdd Vss) KSAdder_Group_PG
    I21 (SG4 SG6 TG6 SP4 SP6 TP6 Vdd Vss) KSAdder_Group_PG
    I20 (SG3 SG5 TG5 SP3 SP5 TP5 Vdd Vss) KSAdder_Group_PG
    I19 (SG2 SG4 TG4 SP2 SP4 TP4 Vdd Vss) KSAdder_Group_PG
    I18 (SG1 SG3 TG3 SP1 SP3 TP3 Vdd Vss) KSAdder_Group_PG
    I16 (G1 G2 SG2 P1 P2 SP2 Vdd Vss) KSAdder_Group_PG
    I15 (G0 G1 SG1 P0 P1 SP1 Vdd Vss) KSAdder_Group_PG
    I17 (SG5 SG7 TG7 SP5 SP7 TP7 Vdd Vss) KSAdder_Group_PG
    I13 (G2 G3 SG3 P2 P3 SP3 Vdd Vss) KSAdder_Group_PG
    I12 (G3 G4 SG4 P3 P4 SP4 Vdd Vss) KSAdder_Group_PG
    I11 (G4 G5 SG5 P4 P5 SP5 Vdd Vss) KSAdder_Group_PG
    I10 (G5 G6 SG6 P5 P6 SP6 Vdd Vss) KSAdder_Group_PG
    I9 (G6 G7 SG7 P6 P7 SP7 Vdd Vss) KSAdder_Group_PG
    I30 (Vss UG7 CarryOut UP7 Vdd Vss) KSAdder_Group_G
    I29 (Vss TG3 UG3 TP3 Vdd Vss) KSAdder_Group_G
    I28 (SG0 TG4 UG4 TP4 Vdd Vss) KSAdder_Group_G
    I27 (TG2 TG6 UG6 TP6 Vdd Vss) KSAdder_Group_G
    I26 (TG1 TG5 UG5 TP5 Vdd Vss) KSAdder_Group_G
    I25 (Vss SG1 TG1 SP1 Vdd Vss) KSAdder_Group_G
    I24 (SG0 SG2 TG2 SP2 Vdd Vss) KSAdder_Group_G
    I23 (Vss G0 SG0 P0 Vdd Vss) KSAdder_Group_G
    I52 (output\<0\> Vdd Vss Vss P0) XOR
    I49 (output\<3\> Vdd Vss P3 TG2) XOR
    I48 (output\<2\> Vdd Vss P2 TG1) XOR
    I47 (output\<5\> Vdd Vss P5 UG4) XOR
```

```
    I50 (output\<4\> Vdd Vss P4 UG3) XOR
    I46 (output\<6\> Vdd Vss P6 UG5) XOR
    I51 (output\<1\> Vdd Vss P1 SG0) XOR
    I31 (output\<7\> Vdd Vss P7 UG6) XOR
ends KoggeStoneAdder
// End of subcircuit definition.

// Library name: DIC
// Cell name: ALU
// View name: schematic
subckt ALU AluOut\<0\> AluOut\<1\> AluOut\<2\> AluOut\<3\> AluOut\<4\> \
        AluOut\<5\> AluOut\<6\> AluOut\<7\> CarryOut control\<0\> \
        control\<1\> control\<2\> inputA\<0\> inputA\<1\> inputA\<2\> \
        inputA\<3\> inputA\<4\> inputA\<5\> inputA\<6\> inputA\<7\> \
        inputB\<0\> inputB\<1\> inputB\<2\> inputB\<3\> inputB\<4\> \
        inputB\<5\> inputB\<6\> inputB\<7\> vdd vss
    OR\<0\> (OrOut\<0\> inputA\<0\> inputB\<0\> vdd vss) OR
    OR\<1\> (OrOut\<1\> inputA\<1\> inputB\<1\> vdd vss) OR
    OR\<2\> (OrOut\<2\> inputA\<2\> inputB\<2\> vdd vss) OR
    OR\<3\> (OrOut\<3\> inputA\<3\> inputB\<3\> vdd vss) OR
    OR\<4\> (OrOut\<4\> inputA\<4\> inputB\<4\> vdd vss) OR
    OR\<5\> (OrOut\<5\> inputA\<5\> inputB\<5\> vdd vss) OR
    OR\<6\> (OrOut\<6\> inputA\<6\> inputB\<6\> vdd vss) OR
    OR\<7\> (OrOut\<7\> inputA\<7\> inputB\<7\> vdd vss) OR
    AND\<0\> (AndOut\<0\> inputA\<0\> inputB\<0\> vdd vss) AND
    AND\<1\> (AndOut\<1\> inputA\<1\> inputB\<1\> vdd vss) AND
    AND\<2\> (AndOut\<2\> inputA\<2\> inputB\<2\> vdd vss) AND
    AND\<3\> (AndOut\<3\> inputA\<3\> inputB\<3\> vdd vss) AND
    AND\<4\> (AndOut\<4\> inputA\<4\> inputB\<4\> vdd vss) AND
    AND\<5\> (AndOut\<5\> inputA\<5\> inputB\<5\> vdd vss) AND
    AND\<6\> (AndOut\<6\> inputA\<6\> inputB\<6\> vdd vss) AND
    AND\<7\> (AndOut\<7\> inputA\<7\> inputB\<7\> vdd vss) AND
    I26 (vss inputA\<0\> inputA\<1\> inputA\<2\> inputA\<3\> inputA\<4\> \
        inputA\<5\> inputA\<6\> inputA\<7\> Inv\<1\> Inv\<2\> Inv\<3\> \
        Inv\<4\> Inv\<5\> Inv\<6\> Inv\<7\> CompOut\<0\> CompOut\<1\> \
        CompOut\<2\> CompOut\<3\> CompOut\<4\> CompOut\<5\> CompOut\<6\> \
        CompOut\<7\> vdd) Complement
    MUX\<0\> (control\<0\> c0\# control\<1\> c1\# control\<2\> c2\# \
        inputB\<0\> adderOut\<0\> CompOut\<0\> shifterOut\<0\> AndOut\<0\> \
        OrOut\<0\> inputA\<0\> comparator\<0\> vss AluOut\<0\> vdd) _sub3
    MUX\<1\> (control\<0\> c0\# control\<1\> c1\# control\<2\> c2\# \
        inputB\<1\> adderOut\<1\> CompOut\<1\> shifterOut\<1\> AndOut\<1\> \
        OrOut\<1\> inputA\<1\> comparator\<1\> vss AluOut\<1\> vdd) _sub3
    MUX\<2\> (control\<0\> c0\# control\<1\> c1\# control\<2\> c2\# \
        inputB\<2\> adderOut\<2\> CompOut\<2\> shifterOut\<2\> AndOut\<2\> \
        OrOut\<2\> inputA\<2\> comparator\<2\> vss AluOut\<2\> vdd) _sub3
    MUX\<3\> (control\<0\> c0\# control\<1\> c1\# control\<2\> c2\# \
        inputB\<3\> adderOut\<3\> CompOut\<3\> shifterOut\<3\> AndOut\<3\> \
        OrOut\<3\> inputA\<3\> comparator\<3\> vss AluOut\<3\> vdd) _sub3
    MUX\<4\> (control\<0\> c0\# control\<1\> c1\# control\<2\> c2\# \
        inputB\<4\> adderOut\<4\> CompOut\<4\> shifterOut\<4\> AndOut\<4\> \
        OrOut\<4\> inputA\<4\> comparator\<4\> vss AluOut\<4\> vdd) _sub3
    MUX\<5\> (control\<0\> c0\# control\<1\> c1\# control\<2\> c2\# \
        inputB\<5\> adderOut\<5\> CompOut\<5\> shifterOut\<5\> AndOut\<5\> \
```

```
        OrOut\<5\> inputA\<5\> comparator\<5\> vss AluOut\<5\> vdd) _sub3
    MUX\<6\> (control\<0\> c0\# control\<1\> c1\# control\<2\> c2\# \
        inputB\<6\> adderOut\<6\> CompOut\<6\> shifterOut\<6\> AndOut\<6\> \
        OrOut\<6\> inputA\<6\> comparator\<6\> vss AluOut\<6\> vdd) _sub3
    MUX\<7\> (control\<0\> c0\# control\<1\> c1\# control\<2\> c2\# \
        inputB\<7\> adderOut\<7\> CompOut\<7\> shifterOut\<7\> AndOut\<7\> \
        OrOut\<7\> inputA\<7\> comparator\<7\> vss AluOut\<7\> vdd) _sub3
    Cap\<0\> (AluOut\<0\> vss) capacitor c=1f
    Cap\<1\> (AluOut\<1\> vss) capacitor c=1f
    Cap\<2\> (AluOut\<2\> vss) capacitor c=1f
    Cap\<3\> (AluOut\<3\> vss) capacitor c=1f
    Cap\<4\> (AluOut\<4\> vss) capacitor c=1f
    Cap\<5\> (AluOut\<5\> vss) capacitor c=1f
    Cap\<6\> (AluOut\<6\> vss) capacitor c=1f
    Cap\<7\> (AluOut\<7\> vss) capacitor c=1f
    I29 (inputB\<0\> InvB\<0\> inputB\<1\> InvB\<1\> inputB\<2\> InvB\<2\> \
        inputA\<0\> inputA\<1\> inputA\<2\> inputA\<3\> inputA\<4\> \
        inputA\<5\> inputA\<6\> inputA\<7\> shifterOut\<0\> \
        shifterOut\<1\> shifterOut\<2\> shifterOut\<3\> shifterOut\<4\> \
        shifterOut\<5\> shifterOut\<6\> shifterOut\<7\> vdd vss vss vss \
        vss vss) Shifter
    I10 (inputA\<0\> inputA\<1\> inputA\<2\> inputA\<3\> inputA\<4\> \
        inputA\<5\> inputA\<6\> inputA\<7\> inputB\<0\> inputB\<1\> \
        inputB\<2\> inputB\<3\> inputB\<4\> inputB\<5\> inputB\<6\> \
        inputB\<7\> vss comparator\<0\> comparator\<1\> comparator\<2\> \
        comparator\<3\> comparator\<4\> comparator\<5\> comparator\<6\> \
        comparator\<7\> vdd) Comparator
    InverterB\<0\> (InvB\<0\> vdd inputB\<0\> vss) Inverter
    InverterB\<1\> (InvB\<1\> vdd inputB\<1\> vss) Inverter
    InverterB\<2\> (InvB\<2\> vdd inputB\<2\> vss) Inverter
    SelectInverter2 (c2\# vdd control\<2\> vss) Inverter
    SelectInverter0 (c0\# vdd control\<0\> vss) Inverter
    SelectInverter1 (c1\# vdd control\<1\> vss) Inverter
    InverterA\<0\> (Inv\<1\> vdd inputA\<1\> vss) Inverter
    InverterA\<1\> (Inv\<2\> vdd inputA\<2\> vss) Inverter
    InverterA\<2\> (Inv\<3\> vdd inputA\<3\> vss) Inverter
    InverterA\<3\> (Inv\<4\> vdd inputA\<4\> vss) Inverter
    InverterA\<4\> (Inv\<5\> vdd inputA\<5\> vss) Inverter
    InverterA\<5\> (Inv\<6\> vdd inputA\<6\> vss) Inverter
    InverterA\<6\> (Inv\<7\> vdd inputA\<7\> vss) Inverter
    I9 (CarryOut vdd vss inputA\<0\> inputA\<1\> inputA\<2\> inputA\<3\> \
        inputA\<4\> inputA\<5\> inputA\<6\> inputA\<7\> inputB\<0\> \
        inputB\<1\> inputB\<2\> inputB\<3\> inputB\<4\> inputB\<5\> \
        inputB\<6\> inputB\<7\> adderOut\<0\> adderOut\<1\> adderOut\<2\> \
        adderOut\<3\> adderOut\<4\> adderOut\<5\> adderOut\<6\> \
        adderOut\<7\>) KoggeStoneAdder
ends ALU
// End of subcircuit definition.

// Library name: DIC
// Cell name: DLatch
// View name: schematic
subckt DLatch Clk Clk\' In Out Vdd Vss
    M5 (net9 Clk net13 Vss) NMOS_VTL w=90n l=50n as=9.45e-15 ad=9.45e-15 \
```

```
        ps=300n pd=300n ld=105n ls=105n m=1
    M0 (In Clk\' net13 Vss) NMOS_VTL w=90n l=50n as=9.45e-15 ad=9.45e-15 \
        ps=300n pd=300n ld=105n ls=105n m=1
    M4 (net13 Clk In Vdd) PMOS_VTL w=180.0n l=50n as=1.89e-14 ad=1.89e-14 \
        ps=390.0n pd=390.0n ld=105n ls=105n m=1
    M6 (net13 Clk\' net9 Vdd) PMOS_VTL w=180.0n l=50n as=1.89e-14 \
        ad=1.89e-14 ps=390.0n pd=390.0n ld=105n ls=105n m=1
    I2 (Out Vdd net13 Vss) Inverter
    I5 (net9 Vdd Out Vss) Inverter
ends DLatch
// End of subcircuit definition.

// Library name: DIC
// Cell name: Register
// View name: schematic
subckt Register Clk IN OUT Vdd Vss
    I2 (Clk Clk\' IN net28 Vdd Vss) DLatch
    I3 (Clk\' Clk net28 OUT Vdd Vss) DLatch
    I4 (Clk\' Vdd Clk Vss) Inverter
ends Register
// End of subcircuit definition.

// Library name: DIC
// Cell name: outputBuffer
// View name: schematic
subckt outputBuffer out vdd vin vss
    M1 (out vin vdd vdd) PMOS_VTL w=180.0woscaler _gpar0 l=50n as=1.89e-14 \
        ad=1.89e-14 ps=390.0n pd=390.0n ld=105n ls=105n m=1
    M0 (out vin vss vss) NMOS_VTL w=90 wsoscaler _gpar0 l=50n as=9.45e-15 \
        ad=9.45e-15 ps=300n pd=300n ld=105n ls=105n m=1
ends outputBuffer
// End of subcircuit definition.

// Library name: DIC
// Cell name: InverterBuffer
// View name: schematic
subckt InverterBuffer out vdd vin vss
    M1 (out vin vdd vdd) PMOS_VTL w=180.0 wiscaler _gpar0 l=50n \
        as=1.89e-14 ad=1.89e-14 ps=390.0n pd=390.0n ld=105n ls=105n m=1
    M0 (out vin vss vss) NMOS_VTL w=90 wiscaler _gpar0 l=50n as=9.45e-15 \
        ad=9.45e-15 ps=300n pd=300n ld=105n ls=105n m=1
ends InverterBuffer
// End of subcircuit definition.

// Library name: DIC
// Cell name: DSProcessorWBInput
// View name: schematic
V2 (excludedvdd! 0) vsource dc=1.0 type=dc
V0 (vdd! net37) vsource dc=pvdd type=dc
R0 (net37 0) resistor r=1
I1 (outALU\<0\> outALU\<1\> outALU\<2\> outALU\<3\> outALU\<4\> \
        outALU\<5\> outALU\<6\> outALU\<7\> CarryOut controlo\<0\> \
        controlo\<1\> controlo\<2\> con\<0\> con\<1\> con\<2\> con\<3\> \
        con\<4\> con\<5\> con\<6\> con\<7\> inputBo\<0\> inputBo\<1\> \
```

```
        inputBo\<2\> inputBo\<3\> inputBo\<4\> inputBo\<5\> inputBo\<6\> \
        inputBo\<7\> vdd! 0) ALU
ControlRegister\<0\> (Clocko control\<0\> controlo\<0\> excludedvdd! 0) \
        Register
ControlRegister\<1\> (Clocko control\<1\> controlo\<1\> excludedvdd! 0) \
        Register
ControlRegister\<2\> (Clocko control\<2\> controlo\<2\> excludedvdd! 0) \
        Register
RegisterB\<0\> (Clocko con1\<0\> outputi\<0\> vdd! 0) Register
RegisterB\<1\> (Clocko con1\<1\> outputi\<1\> vdd! 0) Register
RegisterB\<2\> (Clocko con1\<2\> outputi\<2\> vdd! 0) Register
RegisterB\<3\> (Clocko con1\<3\> outputi\<3\> vdd! 0) Register
RegisterB\<4\> (Clocko con1\<4\> outputi\<4\> vdd! 0) Register
RegisterB\<5\> (Clocko con1\<5\> outputi\<5\> vdd! 0) Register
RegisterB\<6\> (Clocko con1\<6\> outputi\<6\> vdd! 0) Register
RegisterB\<7\> (Clocko con1\<7\> outputi\<7\> vdd! 0) Register
RegisterA\<0\> (Clocko inputAo\<0\> inALU\<0\> vdd! 0) Register
RegisterA\<1\> (Clocko inputAo\<1\> inALU\<1\> vdd! 0) Register
RegisterA\<2\> (Clocko inputAo\<2\> inALU\<2\> vdd! 0) Register
RegisterA\<3\> (Clocko inputAo\<3\> inALU\<3\> vdd! 0) Register
RegisterA\<4\> (Clocko inputAo\<4\> inALU\<4\> vdd! 0) Register
RegisterA\<5\> (Clocko inputAo\<5\> inALU\<5\> vdd! 0) Register
RegisterA\<6\> (Clocko inputAo\<6\> inALU\<6\> vdd! 0) Register
RegisterA\<7\> (Clocko inputAo\<7\> inALU\<7\> vdd! 0) Register
I22\<0\> (net044\<0\> excludedvdd! outputi\<0\> 0) outputBuffer
I22\<1\> (net044\<1\> excludedvdd! outputi\<1\> 0) outputBuffer
I22\<2\> (net044\<2\> excludedvdd! outputi\<2\> 0) outputBuffer
I22\<3\> (net044\<3\> excludedvdd! outputi\<3\> 0) outputBuffer
I22\<4\> (net044\<4\> excludedvdd! outputi\<4\> 0) outputBuffer
I22\<5\> (net044\<5\> excludedvdd! outputi\<5\> 0) outputBuffer
I22\<6\> (net044\<6\> excludedvdd! outputi\<6\> 0) outputBuffer
I22\<7\> (net044\<7\> excludedvdd! outputi\<7\> 0) outputBuffer
I23\<0\> (output\<0\> excludedvdd! net044\<0\> 0) outputBuffer
I23\<1\> (output\<1\> excludedvdd! net044\<1\> 0) outputBuffer
I23\<2\> (output\<2\> excludedvdd! net044\<2\> 0) outputBuffer
I23\<3\> (output\<3\> excludedvdd! net044\<3\> 0) outputBuffer
I23\<4\> (output\<4\> excludedvdd! net044\<4\> 0) outputBuffer
I23\<5\> (output\<5\> excludedvdd! net044\<5\> 0) outputBuffer
I23\<6\> (output\<6\> excludedvdd! net044\<6\> 0) outputBuffer
I23\<7\> (output\<7\> excludedvdd! net044\<7\> 0) outputBuffer
I29\<0\> (con\<0\> excludedvdd! net063\<0\> 0) InverterBuffer
I29\<1\> (con\<1\> excludedvdd! net063\<1\> 0) InverterBuffer
I29\<2\> (con\<2\> excludedvdd! net063\<2\> 0) InverterBuffer
I29\<3\> (con\<3\> excludedvdd! net063\<3\> 0) InverterBuffer
I29\<4\> (con\<4\> excludedvdd! net063\<4\> 0) InverterBuffer
I29\<5\> (con\<5\> excludedvdd! net063\<5\> 0) InverterBuffer
I29\<6\> (con\<6\> excludedvdd! net063\<6\> 0) InverterBuffer
I29\<7\> (con\<7\> excludedvdd! net063\<7\> 0) InverterBuffer
I28\<0\> (con1\<0\> excludedvdd! net059\<0\> 0) InverterBuffer
I28\<1\> (con1\<1\> excludedvdd! net059\<1\> 0) InverterBuffer
I28\<2\> (con1\<2\> excludedvdd! net059\<2\> 0) InverterBuffer
I28\<3\> (con1\<3\> excludedvdd! net059\<3\> 0) InverterBuffer
I28\<4\> (con1\<4\> excludedvdd! net059\<4\> 0) InverterBuffer
I28\<5\> (con1\<5\> excludedvdd! net059\<5\> 0) InverterBuffer
```

```
I28\<6\> (con1\<6\> excludedvdd! net059\<6\> 0) InverterBuffer
I28\<7\> (con1\<7\> excludedvdd! net059\<7\> 0) InverterBuffer
I26\<0\> (net063\<0\> excludedvdd! inALU\<0\> 0) InverterBuffer
I26\<1\> (net063\<1\> excludedvdd! inALU\<1\> 0) InverterBuffer
I26\<2\> (net063\<2\> excludedvdd! inALU\<2\> 0) InverterBuffer
I26\<3\> (net063\<3\> excludedvdd! inALU\<3\> 0) InverterBuffer
I26\<4\> (net063\<4\> excludedvdd! inALU\<4\> 0) InverterBuffer
I26\<5\> (net063\<5\> excludedvdd! inALU\<5\> 0) InverterBuffer
I26\<6\> (net063\<6\> excludedvdd! inALU\<6\> 0) InverterBuffer
I26\<7\> (net063\<7\> excludedvdd! inALU\<7\> 0) InverterBuffer
I27\<0\> (net059\<0\> excludedvdd! outALU\<0\> 0) InverterBuffer
I27\<1\> (net059\<1\> excludedvdd! outALU\<1\> 0) InverterBuffer
I27\<2\> (net059\<2\> excludedvdd! outALU\<2\> 0) InverterBuffer
I27\<3\> (net059\<3\> excludedvdd! outALU\<3\> 0) InverterBuffer
I27\<4\> (net059\<4\> excludedvdd! outALU\<4\> 0) InverterBuffer
I27\<5\> (net059\<5\> excludedvdd! outALU\<5\> 0) InverterBuffer
I27\<6\> (net059\<6\> excludedvdd! outALU\<6\> 0) InverterBuffer
I27\<7\> (net059\<7\> excludedvdd! outALU\<7\> 0) InverterBuffer
I7 (Clocko excludedvdd! net053 0) InverterBuffer
I6 (net053 excludedvdd! Clock 0) InverterBuffer
I14\<0\> (net052\<0\> excludedvdd! inputB\<0\> 0) InverterBuffer
I14\<1\> (net052\<1\> excludedvdd! inputB\<1\> 0) InverterBuffer
I14\<2\> (net052\<2\> excludedvdd! inputB\<2\> 0) InverterBuffer
I14\<3\> (net052\<3\> excludedvdd! inputB\<3\> 0) InverterBuffer
I14\<4\> (net052\<4\> excludedvdd! inputB\<4\> 0) InverterBuffer
I14\<5\> (net052\<5\> excludedvdd! inputB\<5\> 0) InverterBuffer
I14\<6\> (net052\<6\> excludedvdd! inputB\<6\> 0) InverterBuffer
I14\<7\> (net052\<7\> excludedvdd! inputB\<7\> 0) InverterBuffer
I12\<0\> (net048\<0\> excludedvdd! inputA\<0\> 0) InverterBuffer
I12\<1\> (net048\<1\> excludedvdd! inputA\<1\> 0) InverterBuffer
I12\<2\> (net048\<2\> excludedvdd! inputA\<2\> 0) InverterBuffer
I12\<3\> (net048\<3\> excludedvdd! inputA\<3\> 0) InverterBuffer
I12\<4\> (net048\<4\> excludedvdd! inputA\<4\> 0) InverterBuffer
I12\<5\> (net048\<5\> excludedvdd! inputA\<5\> 0) InverterBuffer
I12\<6\> (net048\<6\> excludedvdd! inputA\<6\> 0) InverterBuffer
I12\<7\> (net048\<7\> excludedvdd! inputA\<7\> 0) InverterBuffer
I15\<0\> (inputBo\<0\> excludedvdd! net052\<0\> 0) InverterBuffer
I15\<1\> (inputBo\<1\> excludedvdd! net052\<1\> 0) InverterBuffer
I15\<2\> (inputBo\<2\> excludedvdd! net052\<2\> 0) InverterBuffer
I15\<3\> (inputBo\<3\> excludedvdd! net052\<3\> 0) InverterBuffer
I15\<4\> (inputBo\<4\> excludedvdd! net052\<4\> 0) InverterBuffer
I15\<5\> (inputBo\<5\> excludedvdd! net052\<5\> 0) InverterBuffer
I15\<6\> (inputBo\<6\> excludedvdd! net052\<6\> 0) InverterBuffer
I15\<7\> (inputBo\<7\> excludedvdd! net052\<7\> 0) InverterBuffer
I13\<0\> (inputAo\<0\> excludedvdd! net048\<0\> 0) InverterBuffer
I13\<1\> (inputAo\<1\> excludedvdd! net048\<1\> 0) InverterBuffer
I13\<2\> (inputAo\<2\> excludedvdd! net048\<2\> 0) InverterBuffer
I13\<3\> (inputAo\<3\> excludedvdd! net048\<3\> 0) InverterBuffer
I13\<4\> (inputAo\<4\> excludedvdd! net048\<4\> 0) InverterBuffer
I13\<5\> (inputAo\<5\> excludedvdd! net048\<5\> 0) InverterBuffer
I13\<6\> (inputAo\<6\> excludedvdd! net048\<6\> 0) InverterBuffer
I13\<7\> (inputAo\<7\> excludedvdd! net048\<7\> 0) InverterBuffer
```

```
inA0 ( inputA\<0\> 0 ) vsource type=pulse val0=0 val1=pvdd delay=0
     rise=0.001n fall=0.001n width=pd/2 period=pd
inA1 ( inputA\<1\> 0 ) vsource type=pulse val0=pvdd val1=0 delay=0
     rise=0.001n fall=0.001n width=pd/2 period=pd
inA2 ( inputA\<2\> 0 ) vsource type=pulse val0=0 val1=pvdd delay=0
     rise=0.001n fall=0.001n width=pd/2 period=pd
inA3 ( inputA\<3\> 0 ) vsource type=pulse val0=pvdd val1=0 delay=0
     rise=0.001n fall=0.001n width=pd/2 period=pd
inA4 ( inputA\<4\> 0 ) vsource type=pulse val0=0 val1=pvdd delay=0
     rise=0.001n fall=0.001n width=pd/2 period=pd
inA5 ( inputA\<5\> 0 ) vsource type=pulse val0=pvdd val1=0 delay=0
     rise=0.001n fall=0.001n width=pd/2 period=pd
inA6 ( inputA\<6\> 0 ) vsource type=pulse val0=0 val1=pvdd delay=0
     rise=0.001n fall=0.001n width=pd/2 period=pd
inA7 ( inputA\<7\> 0 ) vsource type=pulse val0=pvdd val1=0 delay=0
     rise=0.001n fall=0.001n width=pd/2 period=pd

inB0 ( inputB\<0\> 0 ) vsource type=pulse val0=pvdd val1=pvdd delay=0
     rise=0.001n fall=0.001n width=pd/2 period=pd
inB1 ( inputB\<1\> 0 ) vsource type=pulse val0=0 val1=0 delay=0
     rise=0.001n fall=0.001n width=pd/2 period=pd
inB2 ( inputB\<2\> 0 ) vsource type=pulse val0=pvdd val1=pvdd delay=0
     rise=0.001n fall=0.001n width=pd/2 period=pd
inB3 ( inputB\<3\> 0 ) vsource type=pulse val0=0 val1=0 delay=0
     rise=0.001n fall=0.001n width=pd/2 period=pd
inB4 ( inputB\<4\> 0 ) vsource type=pulse val0=pvdd val1=pvdd delay=0
     rise=0.001n fall=0.001n width=pd/2 period=pd
inB5 ( inputB\<5\> 0 ) vsource type=pulse val0=0 val1=0 delay=0
     rise=0.001n fall=0.001n width=pd/2 period=pd
inB6 ( inputB\<6\> 0 ) vsource type=pulse val0=pvdd val1=pvdd delay=0
     rise=0.001n fall=0.001n width=pd/2 period=pd
inB7 ( inputB\<7\> 0 ) vsource type=pulse val0=0 val1=0 delay=0
     rise=0.001n fall=0.001n width=pd/2 period=pd

Vcontrol\<0\> ( control\<0\> 0 ) vsource type=pulse val0=0 val1=pvdd
     delay=0 rise=0.001n fall=0.001n width=pd*2 period=pd*4
Vcontrol1\<1\> ( control\<1\> 0 ) vsource type=pulse val0=0 val1=pvdd
     delay=0 rise=0.001n fall=0.001n width=pd*4 period=pd*8
Vcontrol2\<2\> ( control\<2\> 0 ) vsource type=pulse val0=0 val1=pvdd
     delay=0 rise=0.001n fall=0.001n width=pd*8 period=pd*16

ClockSource ( CLK 0 ) vsource type=pulse val0=1 val1=0 delay=pd/10
     rise=0.001n fall=0.001n width=pd/2 period=pd
```